Fluxwire Documentation

Release 3.5.0

Corporation

November 12, 2015

CONTENTS

CHAPTER

ONE

OVERVIEW

Fluxwire is a platform independent and extensible overlay network that provides advanced distributed and mesh networking services. The mesh network is configured using two types of binary applications: Nodes and Controls. Nodes are deployed across a variety of devices and employ interconnects that form the basis of the mesh network (depicted by green icons in figure below). Controls connect to one or more Nodes allowing operators to configure and utilize the mesh network (depicted by blue icons in figure below).

The Fluxwire architecture consists of two components, the core and plug-ins. The core is responsible for distribution of packets, networking, and overall maintenance of the runtime application while plug-ins expand the network with new features. Plug-ins are dynamically loadable and utilize services of the network to provide specialized features and capabilities. Future plug-ins can be added without reengineering the core.

Fluxwire supports the following key features:

- Multiple instances of TCP and Reliable UDP from the same runtime environment.
- Fully autonomous mesh network supporting self-healing, loops, and multi-path routing.
- Extensible with plug-ins, independent of the core application.
- Advanced graphical user interface using proven Java technology.
- Nodes supported across nine major Operating Systems and six different architectures.

1.1 Communications Stack

The core component of Fluxwire utilizes a communication stack, which is designed from the ground-up to handle and scale to a wide range of mesh configurations. The multi-layered stack implements an advanced routing algorithm to support loops and multi-path packet relays.

The following image illustrates the high-level network features of the communications stack.

1.2 Authentication and Encryption

Fluxwire requires authentication for every link established between two endpoints. The Nodes exchange critical session information following successful authentication. In the case of failed authentications, the link is automatically destroyed and the mesh network operators are notified of the unauthorized attempt to access.

Fluxwire supports end-to-end encryption across the entire mesh network. Unlike multi-hop VPN solutions, data is only decrypted once it reaches the destination node. Encryption uses the widely accepted and proven AES algorithm in either 128-bit or 256-bit keys.

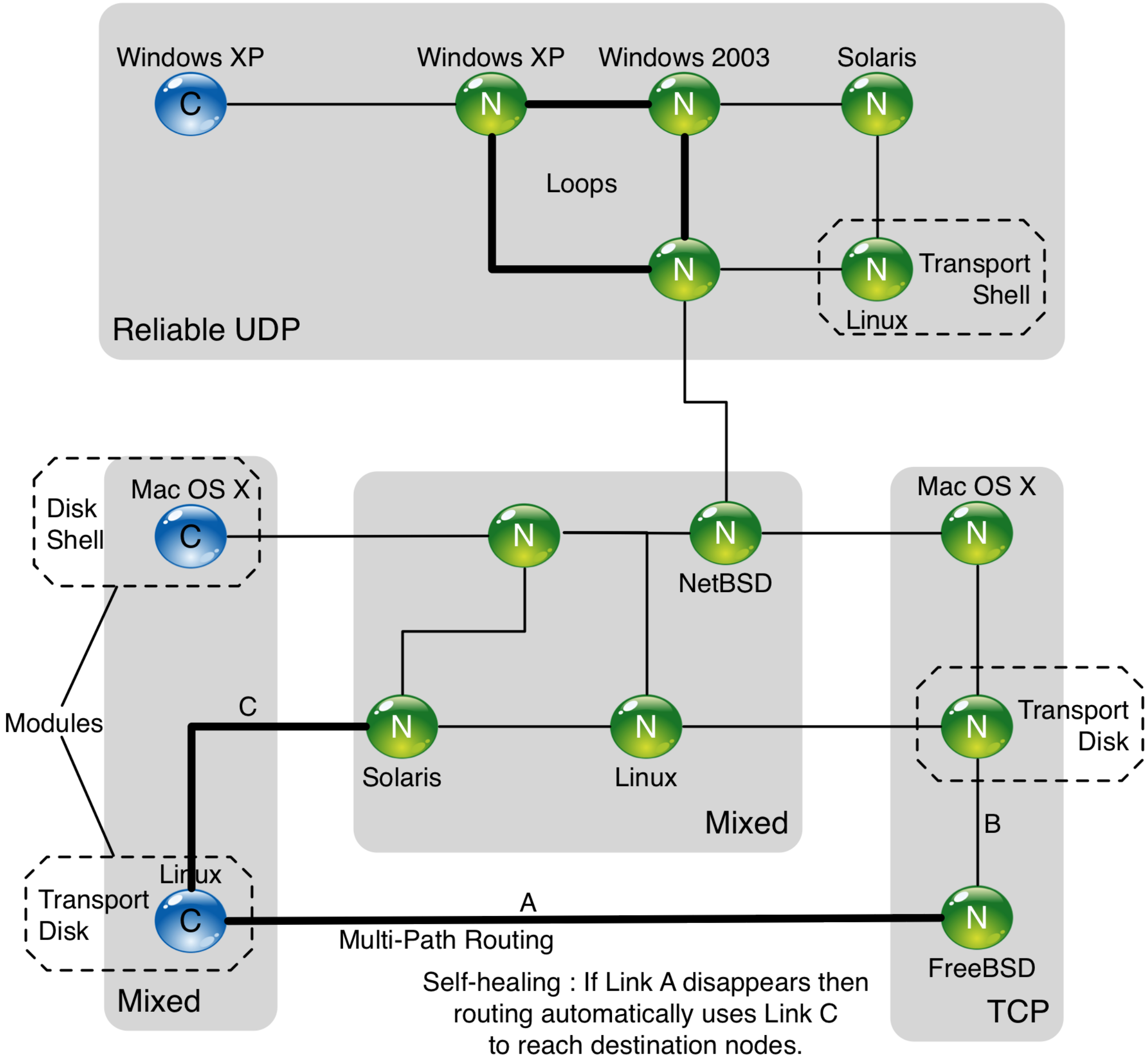
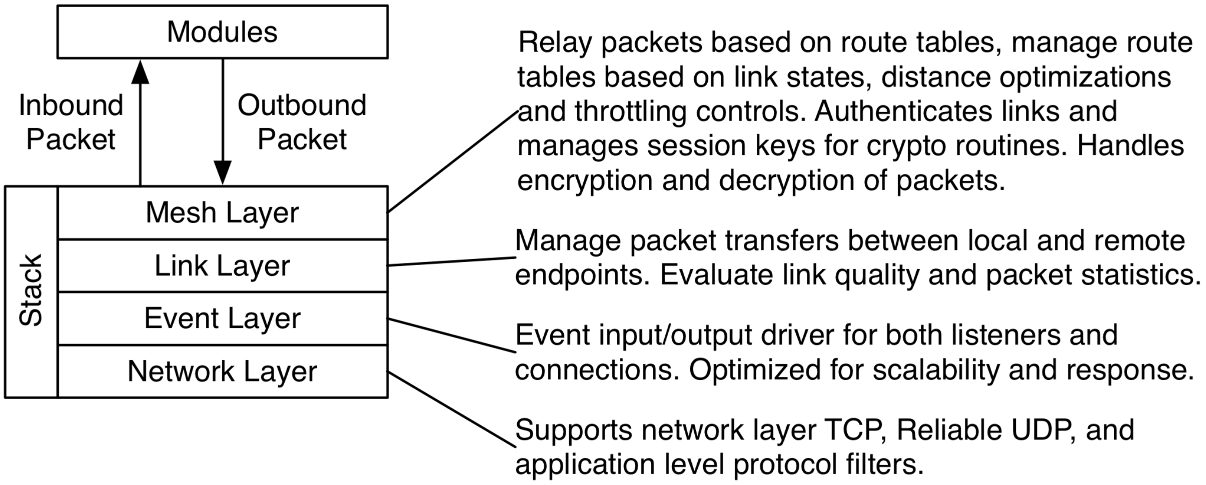


Fig. 1.1: **Mesh Network Deployed Across Numerous Devices**



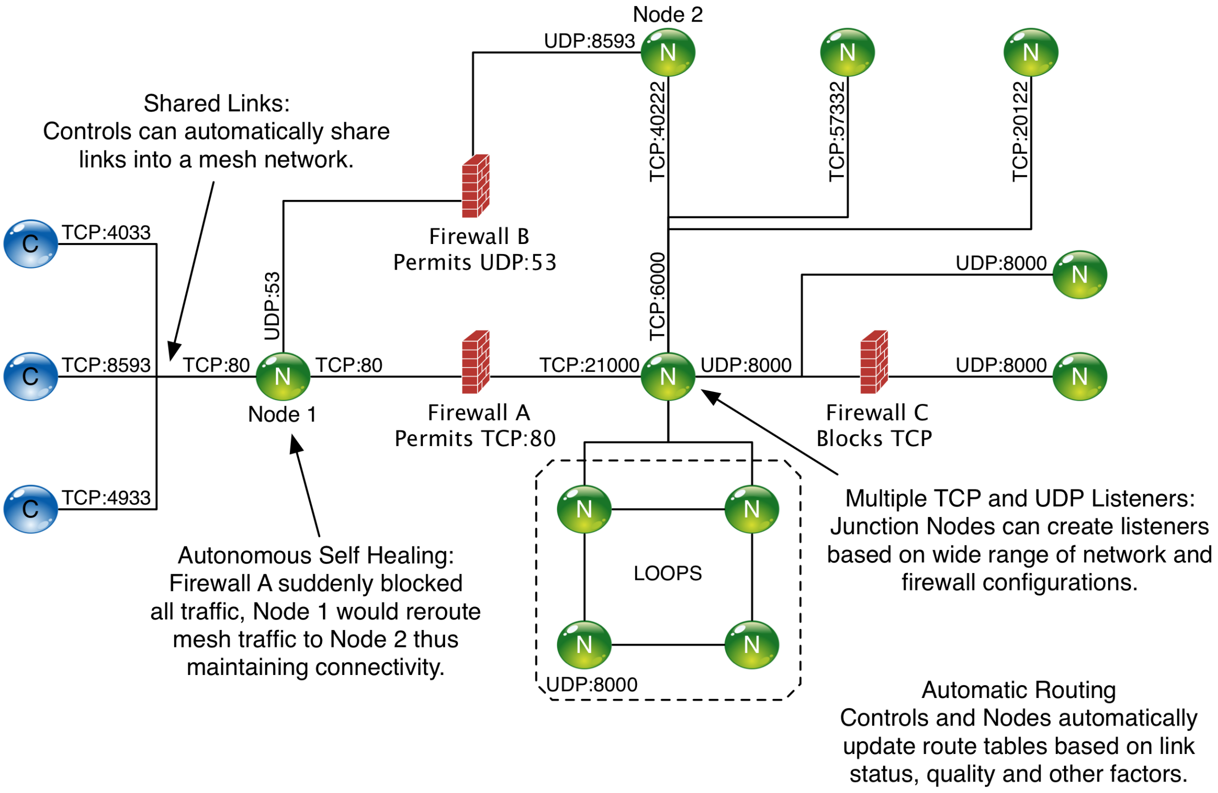
Relay packets based on route tables, manage route tables based on link states, distance optimizations and throttling controls. Authenticates links and manages session keys for crypto routines. Handles encryption and decryption of packets.

Manage packet transfers between local and remote endpoints. Evaluate link quality and packet statistics.

Event input/output driver for both listeners and connections. Optimized for scalability and response.

Supports network layer TCP, Reliable UDP, and application level protocol filters.

Fig. 1.2: **Communications Stack Design**



Shared Links:
Controls can automatically share links into a mesh network.

Autonomous Self Healing:
Firewall A suddenly blocked all traffic, Node 1 would reroute mesh traffic to Node 2 thus maintaining connectivity.

Multiple TCP and UDP Listeners:
Junction Nodes can create listeners based on wide range of network and firewall configurations.

Automatic Routing
Controls and Nodes automatically update route tables based on link status, quality and other factors.

Fig. 1.3: **Mesh Network using the Communications Stack**

1.3 Supported Platforms and Architectures

Platform	x86	x64	PPC	MIPS	ARM	Sparc
Microsoft Windows	✓	✓			✓	
Linux	✓	✓	✓	✓	✓	✓
Mac OS X	✓	✓				
FreeBSD	✓					
OpenBSD						
NetBSD						
Solaris	✓					✓
AIX			✓			
SCO						

✓ = Supported | ⚠ = Experimental

Linux Based Distributions:

- Airosv MIPS-BE
- Ar71xx MIPS-BE
- Backfire ARM-BE
- CPE MIPS-LE
- GP ARM-LE
- ARMV5-LE (DVR)
- Inca MIPS-LE, MIPS-BE
- IXP4xx ARM-BE
- Mikrotik RouterOS 4.x-6.x PPC, MIPS-BE, MIPS-LE, x86, TileGX (32b)
- Paradyne MIPS-LE
- Vanguard ARM-LE

1.4 Nodes

Nodes are deployed across a variety of devices and they employ interconnections that form the basis of the mesh network. The Node is a single binary executable optimized to operate on a specific platform and architecture. Nodes are engineered to be very portable and operate on a wide range of device types: servers, workstations, desktops, laptops, and embedded appliances. The binary format contains all of the functionality associated with runtime configurations, mesh network communications, and plug-in maintenance. Nodes receive and load the plug-ins from the mesh network. Using a Control application, an operator can dynamically upload, activate, and configure more than one plug-in onto every Node.

1.5 Controls

Controls connect to more than one Node and enable operators to configure and utilize the mesh network. A Control is equipped with a plug-in database that stores required plug-ins for Nodes and Controls. The Control loads its own version of plug-ins that includes additional operator centric features. With respect to their application, Controls are divided into two application components: the backend and clients. The backend executes all of the logic and functionality associated with runtime configurations, mesh network communications, and plug-in maintenance. The clients are the operator interface into Fluxwire and connect to the Control backend using a network connection and an open source extensible protocol format. The two Control clients deployed with Fluxwire are command line and graphical user interface.

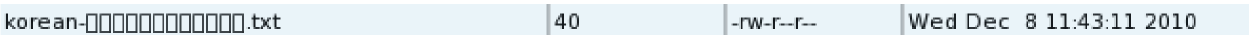
1.6 Graphical User Interface

Operators utilize a Graphical User Interface to issue commands and monitor for near real-time changes in the mesh network. Key features of the graphical user interface client include are:

- Distinct application that communicates with a Control backend using the widely proven message format "Protobufs" developed by Google.
- Based on proven Java technology with custom widget and visual enhancements to provide a responsive and intelligent interface.
- Real-time display of mesh network diagrams and information.
- Real-time validation of commands to prevent operator errors.
- Common interface across three major operating systems: Microsoft Windows, Mac OS X, and Linux.
- Multiple user interfaces that can communicate with a single Control backend.
- Unicode support (UTF-8/UTF-16)

1.7 Language Packs

Fluxwire supports a number of different languages through the implementation of Unicode (UTF-8/UTF-16) support. This allows modules such as Disk and Shell to display special characters and other languages properly in the GUI. The only requirement is that the necessary language packs must be installed on the machine running the GUI. An example of the Korean language pack missing on Fedora Core 10 is shown below.



```
korean-□□□□□□□□□□□□.txt          40          -rw-r--r--          Wed Dec  8 11:43:11 2010
```

Fig. 1.4: **Fedora Core 10 - Missing Korean Language Pack**

1.7.1 Fedora Core 10 - Chinese, Japanese, and Korean (CJK)

```
yum install xorg-x11-xfs
yum erase cjkunifonts-uming
yum erase cjkunifonts-ukai
rpm -i chkfontpath-1.10.1-1.1.i386.rpm
rpm -i fonts-chinese-3.02-4.1.noarch.rpm
yum groupinstall japanese-support
yum groupinstall korean-support
```

Note: Ubuntu 10.04.4 AMD64 (Desktop) does not require that additional language packs be installed.

1.7.2 Ubuntu 12 and Debian 7.8.0 - Chinese, Japanese, and Korean (CJK)

```
apt-get install fonts-arphic-ukai fonts-arphic-uming
apt-get install fonts-ipafont-mincho fonts-ipafont-gothic
apt-get install fonts-unfonts-core fonts-nanum fonts-nanum-coding fonts-nanum-extra fonts-nanum-gothi
```

CHAPTER TWO

APPLICATIONS

Fluxwire is composed of five primary applications:

- Fluxwire Manager (flx-manager): Helper application to manage installations.
- Fluxwire Packer (flx-packer): Helper application for building preconfigured nodes.
- Fluxwire Node: Remote application that forms the basis of the mesh network.
- Fluxwire Desktop (flx-desktop): Control backend application.
- Fluxwire Graphical User Interface Client (flx-gui): Control client application that interfaces with the Desktop.

2.1 Manager (flx-manager)

Fluxwire Manager is a helper application that manages installations and provides access to the default .conf file for the currently selected version. It provides several commands: *List*, *Conf*, *Change*, and *Uninstall*.

View available commands and their options:

```
# flx-manager -h

usage: flx-manager [-h] {list,conf,change,uninstall} ...

Manage installed Fluxwire versions

positional arguments:
  {list,conf,change,uninstall}
    list                List installed version(s)
    conf                Get configuration file for default version
    change              Change default version
    uninstall           Uninstall package(s)

optional arguments:
  -h, --help            show this help message and exit
```

flx-manager <command> -h

```
# flx-manager conf -h

usage: flx-manager conf [-h] [-o OUTPUT]

optional arguments:
  -h, --help            show this help message and exit
  -o OUTPUT, --ouput OUTPUT
                        Output configuration to file
```

2.1.1 List

List all installed versions and where they are located on the system. The * indicates which version is the currently selected version.

```
# flx-manager list

    Version        /usr/local/fluxwire/
   ---------  ------------------------------
*      3.3.0  3.3.0
       3.3.0  3.3.0-rc3
       3.3.0  3.3.0-rc2
       3.3.0  3.3.0-rc1
```

2.1.2 Conf

Display or write to a file the default .conf file for the currently selected version. Shown below are several ways to view/save the .conf file.

```
# flx-manager conf

# Desktop Configuration File

version = "3.3.0";

desktop: {
  /* Name of the desktop instance */
  name = "desktop";

  directory: {
    /* Install directory */
    install = "/usr/local/fluxwire/default";

    /* Base directory for all logs and database storage for current session */
    session = "basedir";
  };

  ...
```

```
# flx-manager conf -o default.conf
```

```
# flx-manager conf > /tmp/default.conf
```

2.1.3 Change

Change the currently selected version to another version. If the ID of the currently selected version is selected no change will be made.

```
# flx-manager change

    ID    Version        /usr/local/fluxwire/
   ----  ---------  ------------------------------
*     1      3.3.0  3.3.0
      2      3.3.0  3.3.0-rc3
      3      3.3.0  3.3.0-rc2
      4      3.3.0  3.3.0-rc1
```

```
Enter new default version [1]: 2
Default updated: 3.3.0-rc3 (3.3.0)

# flx-manager list

    ID    Version        /usr/local/fluxwire/
   ----  ---------  ------------------------------
      1      3.3.0  3.3.0
*     2      3.3.0  3.3.0-rc3
      3      3.3.0  3.3.0-rc2
      4      3.3.0  3.3.0-rc1
```

2.1.4 Uninstall

Uninstall one or all installed Fluxwire versions. The default version cannot be uninstalled unless the ALL option is selected or the default version is changed.

```
# flx-manager uninstall

    ID    Version        /usr/local/fluxwire/
   ----  ---------  ------------------------------
      0             ALL
*     1      3.3.0  3.3.0
      2      3.3.0  3.3.0-rc3
      3      3.3.0  3.3.0-rc2
      4      3.3.0  3.3.0-rc1

Enter ID to uninstall: 4
Uninstalled /usr/local/fluxwire/3.3.0-rc1

# flx-manager list

    ID    Version        /usr/local/fluxwire/
   ----  ---------  ------------------------------
      1      3.3.0  3.3.0
*     2      3.3.0  3.3.0-rc3
      3      3.3.0  3.3.0-rc2
```

2.2 Packer (flx-packer)

Fluxwire Packer is a helper application that generates a Node binary of a specific type: platform and architecture. A preconfigured Node embeds the user’s node configuration variables into an encrypted container of the binary. The executable filename for this application is flx-packer.

```
$ flx-packer
required arguments:
  -o, --output       Archive and self extracting executable name
  -s, --system       System in the format: os:cpu<:model><:version>. Display
                     available systems: -s ?
  -k, --network-key  Network authentication key in 16 alphanumeric characters

one of the following arguments is required:
  --service service  Service configuration
  --link link        Link configuration
```

```
optional arguments:
  -n, --name          Device name
  -m, --mtu           Maximum transmission unit
  -b, --blacklist     List of comma separated plugins to be blacklisted
  -i, --timeout       Timeout in seconds to shutdown node, default 1200:30:on
  -r, --runtime       Can be "file" or "embed"
  -t, --type          Can be "memory" or "disk"
  -l, --library       Executable is a shared object (Windows only)
  -g, --gen-keys      Generate public/private keys
  -e, --extract       Displays the contents of an archive file
  -v, --version       Displays the Fluxwire version
  -h, --help          Displays more help information
  -d, --developer     Developer mode using plugins.conf in given path
  --ssl               Can be "openssl" or "polarssl" (Posix only)
```

Man pages can also be accessed through the following commands:

- flx-packer –help
- flx-packer –help packer
- flx-packer –help service
- flx-packer –help link
- flx-packer –help http
- flx-packer –help https
- flx-packer –help dns
- flx-packer –help rtp
- flx-packer –help ssl

2.2.1 Examples

Below are examples that display how the flx-packer can be used to build nodes.

Linux x86 node with a TCP service port and timeout with delete, overriding the default timeout value.

```
$ flx-packer -o node-lin -s linux:x86 -k aaaabbbbccccdddd \
  -i 600:30:on --service "tcp=5001" -m 1300 -n node-1
```

Linux x86 node with a TCP service port

```
$ flx-packer -o node-lin -s linux:x86 -k aaaabbbbccccdddd \
  --service "tcp=5001" -m 1300 -n node-1
```

Linux x86 node with a TCP service port and hostname as device name

```
$ flx-packer -o node-lin -s linux:x86 -k aaaabbbbccccdddd \
  --service "tcp=5001" -m 1300
```

Microsoft Windows x86 node with a UDP service port and TCP link with local and remote ports defined.

```
$ flx-packer -o node-win.exe -s mswin:x86 -k aaaabbbbccccdddd \
  -m 1300 -n node-1 \
  --service "udp=5002" --link "ip=127.0.0.1 tcp=5030:5001"
```

Linux x86 node with TCP/HTTP link

```
$ flx-packer -o node-lin -s linux:x86 -k aaaabbbbccccdddd \
  -m 1300 -n node-1 \
  --link "ip=127.0.0.12 tcp=80 proto=http depth=5 host=www.test.com"
```

Microsoft Windows x86 node with TCP/HTTP link through a proxy and TCP/HTTP service port with the specified http response header index

```
$ flx-packer -o node-win.exe -s mswin:x86 -k aaaabbbbccccdddd \
  -m 1300 -n node-1 \
  --link "ip=192.168.40.12 tcp=80 proto=http pool=8 proxy=1 depth=5 host=www.test.com"
  --service "tcp=5005 proto=http header=3"
```

Microsoft Windows x86 node with TCP/HTTP link through a proxy (192.168.4.12 on 3128) to www.test.com:8080

```
$ flx-packer -o node-win.exe -s mswin:x86 -k aaaabbbbccccdddd \
  -m 1300 -n node-1 \
  --link "ip=192.168.4.12 tcp=3128 proto=http proxy=1 depth=10 pool=8 host=www.test.com host_port=808
```

Linux x86 node with TCP/HTTPS service port and link

```
$ flx-packer -o node-lin -s linux:x86 -k aaaabbbbccccdddd \
  -m 1300 -n node-1 \
  --service "tcp=5005 proto=https public=/tmp/public.pem private=/tmp/private.pem" \
  --link "ip=127.0.0.1 tcp=5001 proto=https"
```

Linux x86 node with UDP/RTP link

```
$ flx-packer -o node-lin -s linux:x86 -k aaaabbbbccccdddd \
  -m 1300 -n node-1 \
  --link "ip=127.0.0.1 udp=5001 proto=rtp"
```

Linux x86 node with UDP/DNS link

```
$ flx-packer -o node-lin -s linux:x86 -k aaaabbbbccccdddd \
  -m 1300 -n node-1 \
  --link "ip=127.0.0.1 udp=5001 proto=dns domain=test1.test.com"
```

Linux x86 node with TCP/SSL service (defaults to using OpenSSL on remote system)

```
$ flx-packer -o node-lin -m 1300 -s linux:x86  -k aaaabbbbccccdddd \
  --service "tcp=5001 proto=ssl public=/public.pem private=/private.pem"
```

Linux x86 node with TCP/SSL link (defaults to using OpenSSL on remote system)

```
$ flx-packer -m 1300 -s linux:x86 -o node-lin -k aaaabbbbccccdddd \
  --link "ip=127.0.0.1 tcp=5001 proto=ssl"
```

Linux x86 node with TCP/SSL service using OpenSSL library installed on the remote system

```
$ flx-packer -o node-lin -m 1300 -s linux:x86 --ssl openssl \
  -k aaaabbbbccccdddd --service "tcp=5001 proto=ssl public=/public.pem private=/private.pem"
```

Linux x86 node with TCP/SSL service using Fluxwire’s PolarSSL library

```
$ flx-packer -o node-lin -m 1300 -s linux:x86 --ssl polarssl \
  -k aaaabbbbccccdddd --service "tcp=5001 proto=ssl public=/public.pem private=/private.pem"
```

flx-packer IP6 Service and Link Examples

IP version key can take values ‘4’, ‘6’, ‘4,6’, or ‘6,4’. See *IP Version 6 Support* for details.

Note: 4 is the default IP version if ipv is not specified. Thus, all the examples above are IP version 4 only.

Linux x64 node with a TCP service port for IP6 only. No IP4 links will be accepted on 5001 with the below.

```
$ flx-packer -o node-lin -s linux:x64 -k aaaabbbbccccdddd \
  --service "tcp=5001 ipv=6" -m 1300 -n node-1
```

Linux x64 node with a TCP service port for IP4 or IP6. Both IP4 and IP6 links will be accepted on 5001 with the below.

```
$ flx-packer -o node-lin -s linux:x64 -k aaaabbbbccccdddd \
  --service "tcp=5001 ipv=dual" -m 1300 -n node-1
```

Linux x64 node with a UDP IP6 link to the IP6 address specified.

```
$ flx-packer -o node-lin -s linux:x64 -k aaaabbbbccccdddd \
  -m 1300 -n node-1 \
  --link "ip=fd5d:31e1:a36b:a25b::25 udp=5001 ipv=6"
```

Linux x64 node with a TCP link to the IP address resolved for the hostname "alpha.my-example.lan". The IP Version 4,6 means it prefers to resolve the hostname with IP4. If the machine cannot resolve with IP4 then it will try to resolve to an IP6 address. A link request is made only once, for the first successfully resolved IP4 or IP6 address though. See *IP Version 6 Support* for a detailed discussion.

```
$ flx-packer -o node-lin -s linux:x64 -k aaaabbbbccccdddd \
  -m 1300 -n node-1 \
  --link "ip=alpha.my-example.lan tcp=5001 ipv=4,6"
```

Similar to the previous example, but with IP Version preference to IP6 before IP4 for hostname resolution.

```
$ flx-packer -o node-lin -s linux:x64 -k aaaabbbbccccdddd \
  -m 1300 -n node-1 \
  --link "ip=alpha.my-example.lan tcp=5001 ipv=6,4"
```

2.3 Node

Fluxwire Node is the remote application that establishes the interconnecting links of the mesh network. The packer is used to configure and generate nodes for use across multiple platforms. Operators do not interface directly with a node but rather establish a link to the underlying mesh network using the Desktop.

Warning: Microsoft Windows binaries that are located on a network share should be spawned with cmd.exe or CreateProcess(). This is to avoid having a popup dialog appear on the desktop asking the user if they would like to run the file.

2.4 Desktop (flx-desktop)

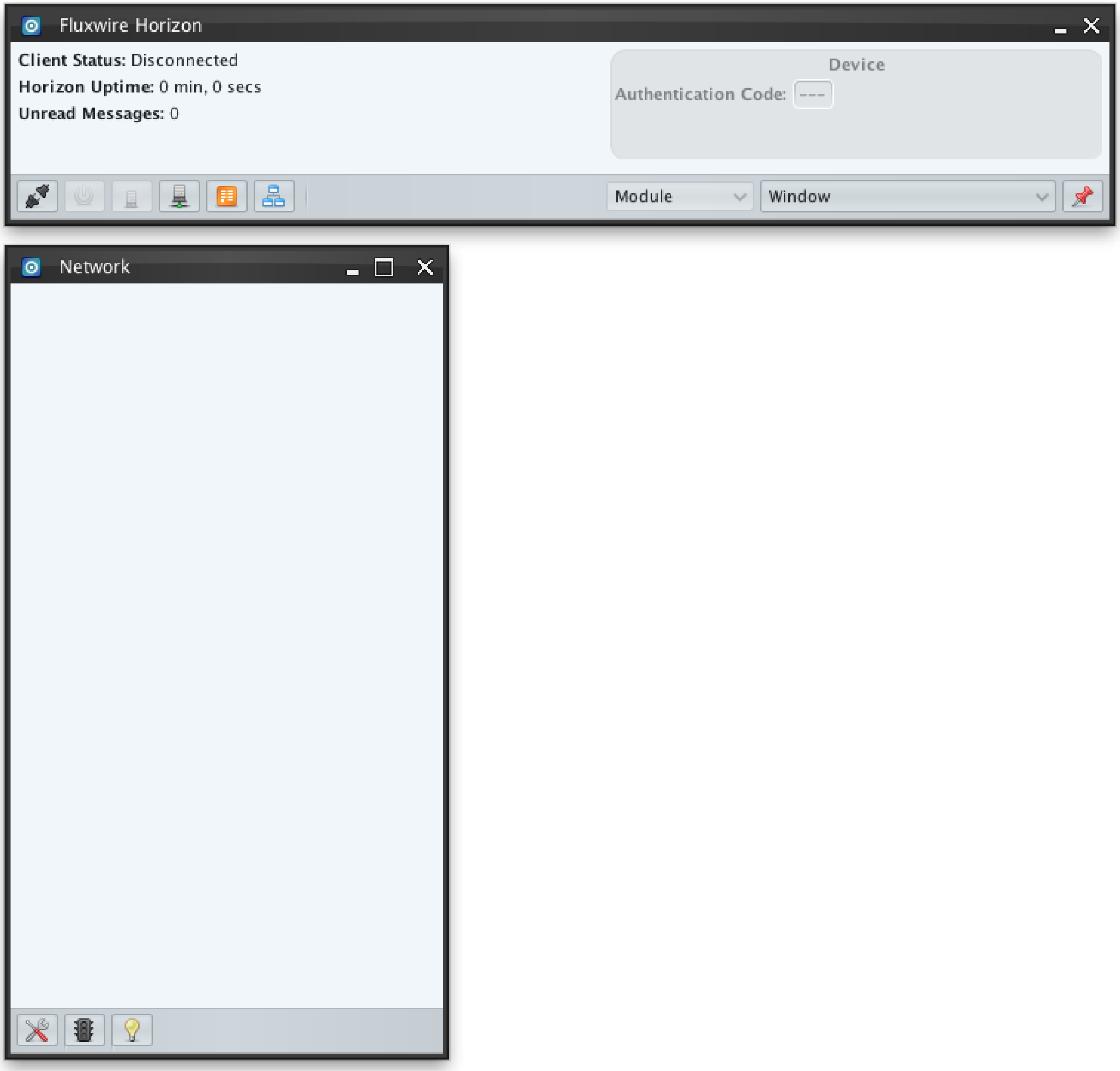
Fluxwire Desktop is the Control application that directly interfaces with the mesh network. An operator does not command or control the Desktop directly but rather utilizes one or more user interface clients. The executable filename for this application is flx-desktop. The application should be run locally in a secure environment where it can receive commands and respond with updates to all clients. The Desktop is initially configured through a mandatory configuration file. A sample configuration file is provided in the following section: *Configuration File*

```
# flx-desktop linux-fluxwire.conf
```

2.5 Graphical User Interface (flx-gui)

Fluxwire Graphical User Interface Client provides an operator interface for mesh networking services and facilities. The executable filename for this application is flx-gui. This application depends on the Java language and requires a virtual machine environment to be present on the local workstation (Only Sun's JVM version 6 is recommended and all other versions are currently unsupported).

```
# flx-gui
```



Fluxwire Horizon

Client Status: Disconnected
Horizon Uptime: 0 min, 0 secs
Unread Messages: 0

Device
Authentication Code: ---

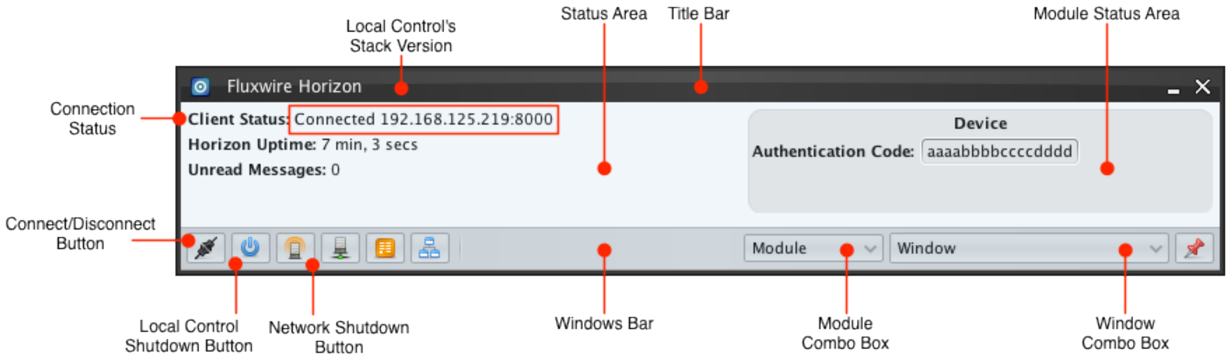
Fig. 2.1: **Fluxwire Layout**

CHAPTER
THREE

DASHBOARD

The Dashboard is the main application window for the flx-gui client and is always visible. The Dashboard enables connecting to Controls (flx-desktop); once connected to a Control, the Control is referred to as the client's *Local Control*.

Dashboard serves as the central command and control window for all of the client's windows. The Dashboard is divided into two sections: Status Area and Window Bar. The Status Area provides statistics and status updates on the overall connection to client's Local Control. The Windows Bar provides buttons for window management of all client windows along with the "Connection" button to connect and disconnect from Controls.



Client Status: Connected 192.168.125.219:8000
Horizon Uptime: 7 min, 3 secs
Unread Messages: 0

Authentication Code: aaaabbbbccccdddd

Fig. 3.1: **Connected Dashboard Window**

3.1 Exiting the flx-gui Client

The Dashboard's close button in the Title Bar quits the flx-gui application. If the flx-gui is currently connected it disconnects and exits.

3.2 Connection Management

flx-gui is configured using a connection to a Control, which becomes the client's Local Control.

The "Connection" button connects and disconnects from a Control. The button is the left button in the Windows Bar highlighted in *Connected Dashboard Window* above. The button toggles its icon to represent its current state.

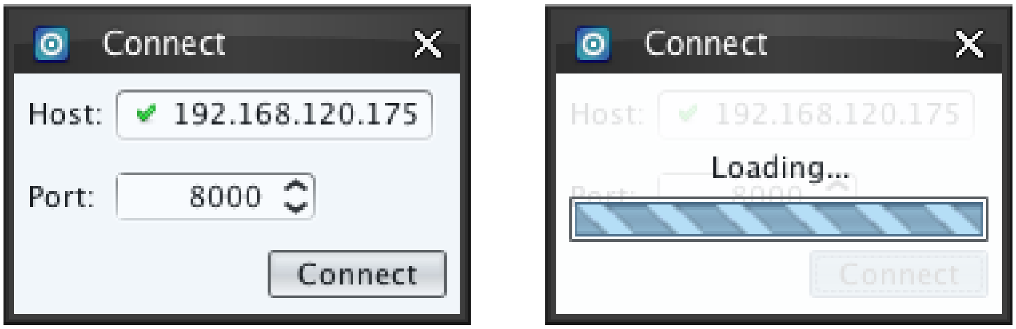
An operator can configure a communication channel to a Control. The disconnected plug icon provides a quick status indicator, signaling that flx-gui has no connection or lost its connection.

🔌 An operator can disconnect the communication channel from the Local Control. The connected plug provides a quick status indicator, signaling that flx-gui has an established connection to a Control.

3.2.1 Connecting

The flx-gui starts disconnected. Clicking the "Connection" button 🔌 makes the Connect Dialog appears. The configuration requires two parameters:

- Host: The IP address of the Control application. Normally the client operates on the same workstation as the flx-desktop Control application and defaults to a loopback address, 127.0.0.1.
- Port: The default configuration file determines the network port of the Control.



Connect

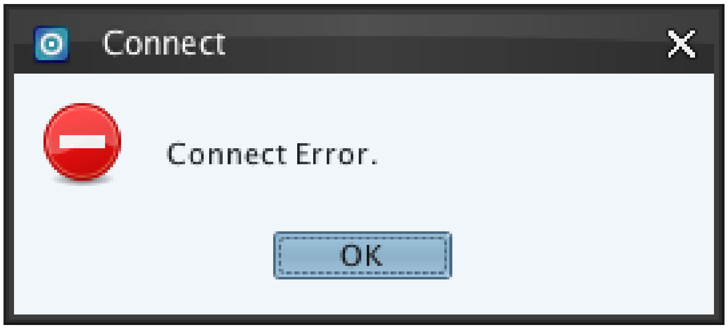
Host: 192.168.120.175

Port: 8000

Fig. 3.2: **Connect Dialog/Loading**

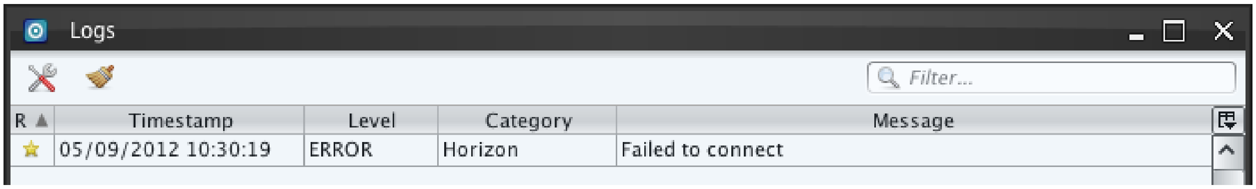
When an operator clicks the "Connect" button in the Connect Dialog and the flx-gui client attempts to connect to a Control. If the connection fails, a "Connection Error" dialog appears along with a new log entry in the *Logs Window*.

If the connection is successful the "Connection" button on the Dashboard changes to 🔌 and the "Client Status" label changes to "Connected <Server Address>:<Port>" and the Local Control's stack version displays in the titlebar as shown in *Connected Dashboard Window*. Other windows also begin populating as the flx-gui is synchronized with the Local Control.



Connect Error.

Fig. 3.3: **Connect Error Message Box**

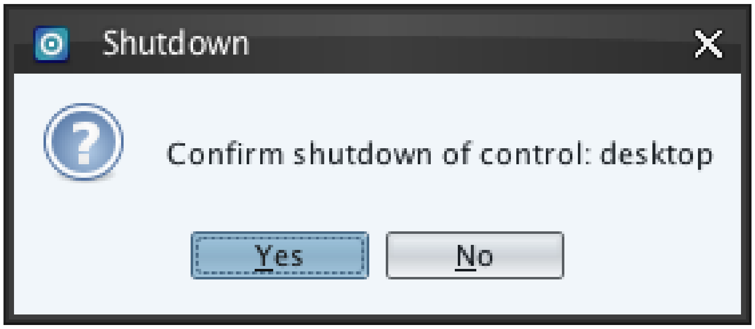


R	Timestamp	Level	Category	Message
☆	05/09/2012 10:30:19	ERROR	Horizon	Failed to connect

Fig. 3.4: **Connect Error Log Entry**

Local Shutdown

The local control can be shutdown by clicking on the power button . A confirmation dialog will be presented to the operator to verify the action performed.



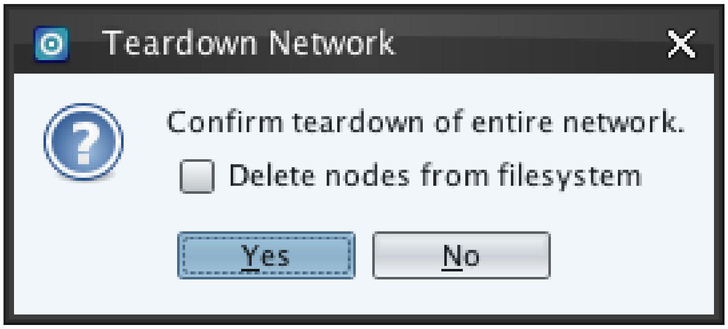
Confirm shutdown of control: desktop

Fig. 3.5: **Local Shutdown Confirmation Dialog Box**

Network Shutdown

The entire network can be shutdown by clicking on the network shutdown button . A confirmation dialog will be presented to the operator to verify the action performed. Nodes can also be deleted by selecting the "Delete nodes from filesystem" checkbox. If the "Yes" button is clicked in the confirmation dialog the shutdown process will begin, starting with the outermost node(s) in the network. This process will continue until only the local control is left in the Network Window or an operator clicks the cancel shutdown button in the Dashboard.

Note: Controls are not instructed to shutdown since other operators may be managing them with flx-gui.



Confirm teardown of entire network.

☐ Delete nodes from filesystem

Fig. 3.6: **Network Shutdown Confirmation Dialog Box**

3.2.2 Disconnecting

Operators disconnect by pressing the "Connection" button while in the connected state. Upon disconnecting the following events transpire.

1. The Logs Window receives a message indicating the client is disconnecting
2. The Network Window clears
3. The Visual Map clears
4. All module windows that were open become disabled
5. The Dashboard's "Client Status" changes to "Disconnected."

3.3 Status Area

The Status Area provides several labels to track the current connection to the Local Control.

- Client Status: Control connection state - "Connected" or "Disconnected"
- Horizon Uptime: Total time the flx-gui has been running.
- Unread Messages: This is a numeric count indicating the number of messages in the Logs Window that require operator acknowledgement. See *Dashboard's Logging Components*.

3.4 Module Status Area

Each module can display critical information to an operator in this central location. To view a different module an operator must click in the Module Status Area. Modules that are loaded on the Local Control are denoted by dark text in the Module Status Area while unloaded modules are denoted by a lighter color text.

3.5 Windows Bar

The Windows Bar contains two groups of components. The left-side contains static buttons for bringing up several common client windows. The right-side contains combo boxes which are dynamically populated based on the client's state and visible windows. pins the Dashboard on top of all windows on the workstation's desktop.

3.5.1 Static Window Components

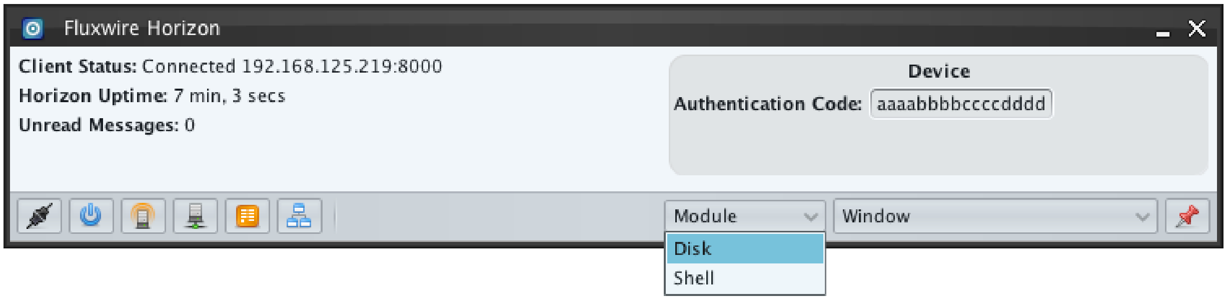
The "Network Window" button displays the *Network Window* or brings the window to the front if it is already visible.

The "Logs Window" button displays the *Logs Window* or brings the window to the front if it is already visible.

The "Visual Map" button displays the *Visual Map Window* or brings it to the front if it is already visible.

3.5.2 Dynamic Window Components

Each Module can provide one optional global window. Global windows are not associated with any one module asset, rather they contain functionality applicable to the whole module like user interface preferences or asset tracking tables. The "Modules" combo box on Dashboard lists all global windows for any module currently enabled on the Local Control. For example, in the figure below the Local Control has "Disk" and "Shell" loaded while "Transport" is unloaded and not visible.



Client Status: Connected 192.168.125.219:8000

Horizon Uptime: 7 min, 3 secs

Unread Messages: 0

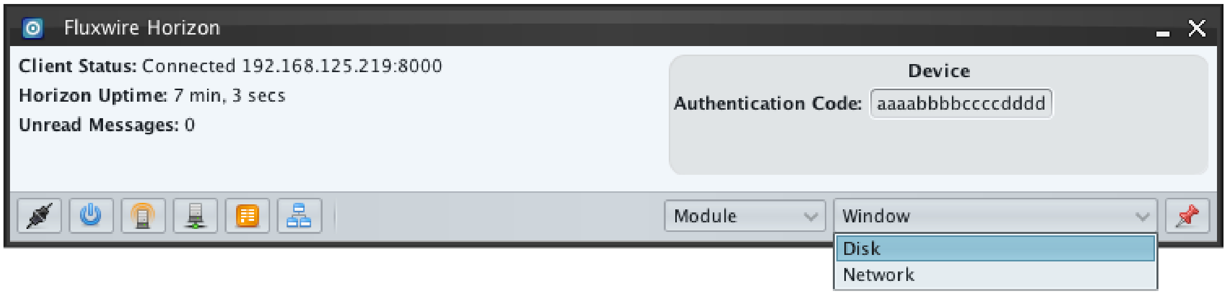
Device

Authentication Code: aaaabbbbccccdddd

Fig. 3.7: **Dashboard Module Menu**

Global windows are uniformly formatted with a series of panels, but the number of panels and their contents are specific to each module. The universal format includes the title bar stating only the module's name, such as "Shell" or "Disk", a side panel with one or more options, and a main content area that changes based on the option selected in the side panel.

The "Windows" combo box lists the titles of all the currently visible windows. Selecting a window in the "Windows" combo box causes that window to be brought to front.



Client Status: Connected 192.168.125.219:8000

Horizon Uptime: 7 min, 3 secs

Unread Messages: 0

Device

Authentication Code: aaaabbbbccccdddd

Fig. 3.8: **Dashboard Windows Menu**

3.5.3 Keep on Top Button

The "Keep on Top" button is a toggle button that enables and disables keeping the Dashboard window on top of all other windows on the workstation's desktop. When this button is enabled only the focus window, the window receiving mouse and keyboard input, can appear on top of the Dashboard window.

CHAPTER

FOUR

LOGGER

The flx-gui client includes a rolling event logging feature consisting of rolling log file(s) and a graphical Log Window. Rolling Log files are placed in the directory where the flx-gui is launched. There are five event logging levels ordered in severity shown in the table below. Each Fluxwire module has its own logging category along with a "Horizon" category for messages relating to flx-gui's connection to its Local Control.

Table 4.1: Logging Levels

Level	Usage
Debug	Debugging events. This level creates a larger number of messages that most operators will not require, but are useful for in depth troubleshooting.
Info	Informative events. Events sent to this level indicate a specific activity such as a file transfer status coming from the Disk module or a Node being shutdown.
Warn	Warning events. Events sent to this level indicate unexpected recoverable behaviors.
Error	Error events. Events sent to this level indicate unexpected behavior that may have unrecoverable side effects in the client. A client's network connection to their Local Control being severed is an example of an error.

Table 4.2: Logging Categories

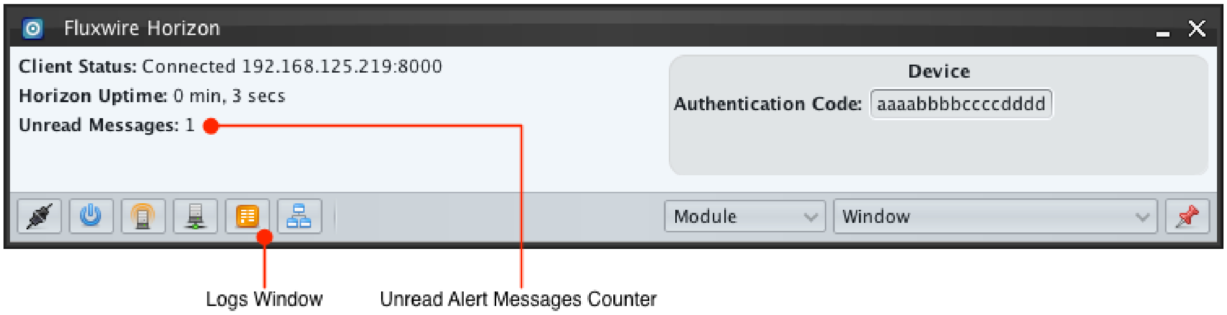
Category	Purpose
Horizon	Messages relating to the flx-gui client, specifically the network connection to the local control.
Chat	Chat Module
Device	Device Module
Disk	Disk Module
Loader	Loader Module
Shell	Shell Module
Transport	Transport Module

4.1 Logs Window

The Logs Window displays logging messages in a tabular format.

Clicking the "Logs Window" button on the Dashboard to display the Logs Window. When the Logs Window is already visible it is brought to front.

The Log Window includes two buttons in the Tool Bar.



Fluxwire Horizon

Client Status: Connected 192.168.125.219:8000

Horizon Uptime: 0 min, 3 secs

Unread Messages: 1

Device

Authentication Code: aaaabbbbccccdddd

Fig. 4.1: **Dashboard's Logging Components**

The "Clear Logs" button deletes all messages in the Logs Window.

Note: Any unread alert messages present when the "Clear Logs" button is clicked are automatically marked as read.

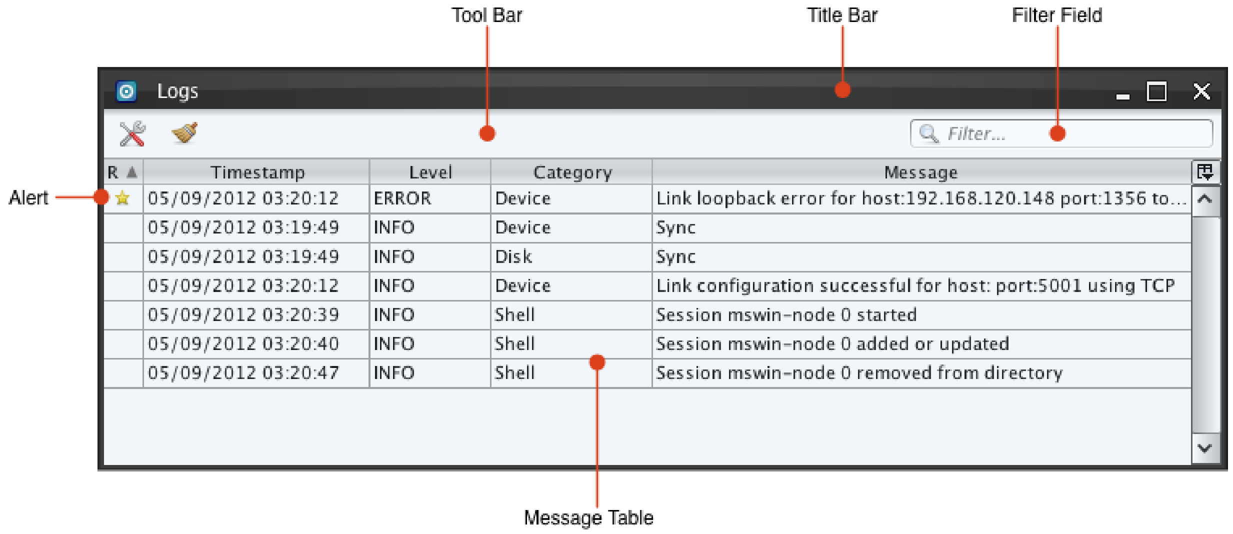
The "Options Dialog" button brings up the Logging Options Dialog box to allow filtering of messages in the Message Table by log level or categories. See *Log Options Dialog*.

Each message is added to the top of the log window so it appears in chronological order from the Local Control's perspective. A scrollbar on the right side allows users to scroll through logging table entries. Some messages have an alert status, which requires the operator to acknowledge the message, however, most messages do not. When an alert is received and the Logs Window is not visible it is immediately made visible. The Message Table contains columns as described below.

- R *(Read Status)*: This column contains a star for an alerting message requesting operator acknowledgement. The operator clears the alert by reading the message and clicking on the row in the Logs Window. Upon acknowledgement, the star is removed. Finally, the Dashboard displays an "Unread Messages" counter that indicates the number of unread alerts. See *Dashboard's Logging Components*.
- Timestamp: Displays the date and time the message was logged.
- Level: Indicates the logging level of the message. See *Logging Levels*.
- Category: The module (or Horizon) that generated the message. See *Logging Categories*.
- Message: The event or message being logged.

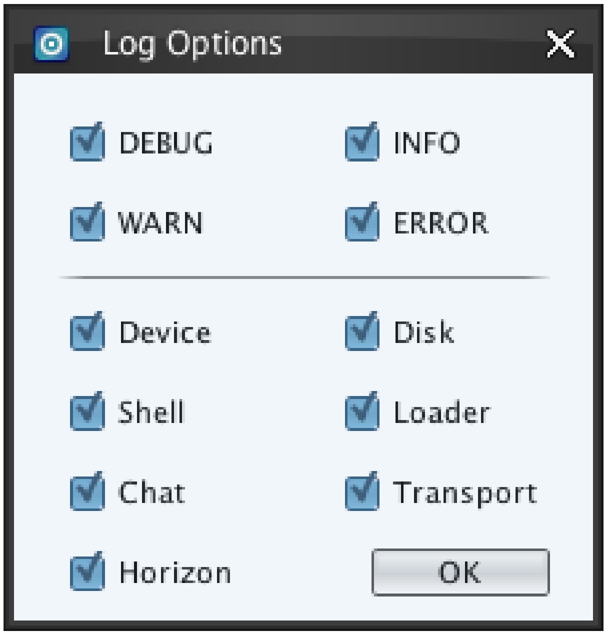
4.2 Log Options Dialog

The Log Options Dialog box allows filtering of an individual log level or category. Log levels display above the horizontal bar in the dialog box and categories display below. By default all logs levels and categories are selected, which makes all messages visible in the Message Table. Clicking the "OK" button exits the dialog and applies the selected filter.



R	Timestamp	Level	Category	Message
☆	05/09/2012 03:20:12	ERROR	Device	Link loopback error for host:192.168.120.148 port:1356 to…
	05/09/2012 03:19:49	INFO	Device	Sync
	05/09/2012 03:19:49	INFO	Disk	Sync
	05/09/2012 03:20:12	INFO	Device	Link configuration successful for host: port:5001 using TCP
	05/09/2012 03:20:39	INFO	Shell	Session mswin-node 0 started
	05/09/2012 03:20:40	INFO	Shell	Session mswin-node 0 added or updated
	05/09/2012 03:20:47	INFO	Shell	Session mswin-node 0 removed from directory

Fig. 4.2: **Logs Window**



☑ DEBUG ☑ INFO

☑ WARN ☑ ERROR

☑ Device ☑ Disk

☑ Shell ☑ Loader

☑ Chat ☑ Transport

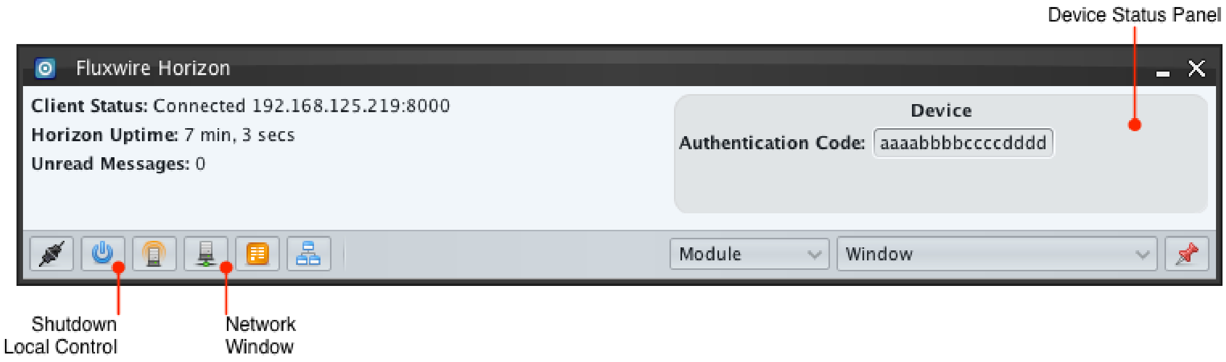
☑ Horizon

CHAPTER

FIVE

DEVICE MODULE

The device module is responsible for all device related activities ranging from the configuration to maintenance and diagnostics of the mesh network. Mesh network devices are composed of Controls and Nodes, all of which interconnect to provide a fully autonomous mesh network. Each device can either wait for an incoming connection using a service port or initiate a connection to ultimately establish a link. The device module has five windows: *Network Window*, *Diagnostics Window*, *Device Window*, *Device Configuration Window*, and *Visual Map Window*.



Fluxwire Horizon

Client Status: Connected 192.168.125.219:8000
Horizon Uptime: 7 min, 3 secs
Unread Messages: 0

Device
Authentication Code: aaaabbbbccccdddd

Fig. 5.1: **Dashboard**

5.1 Device Status Panel

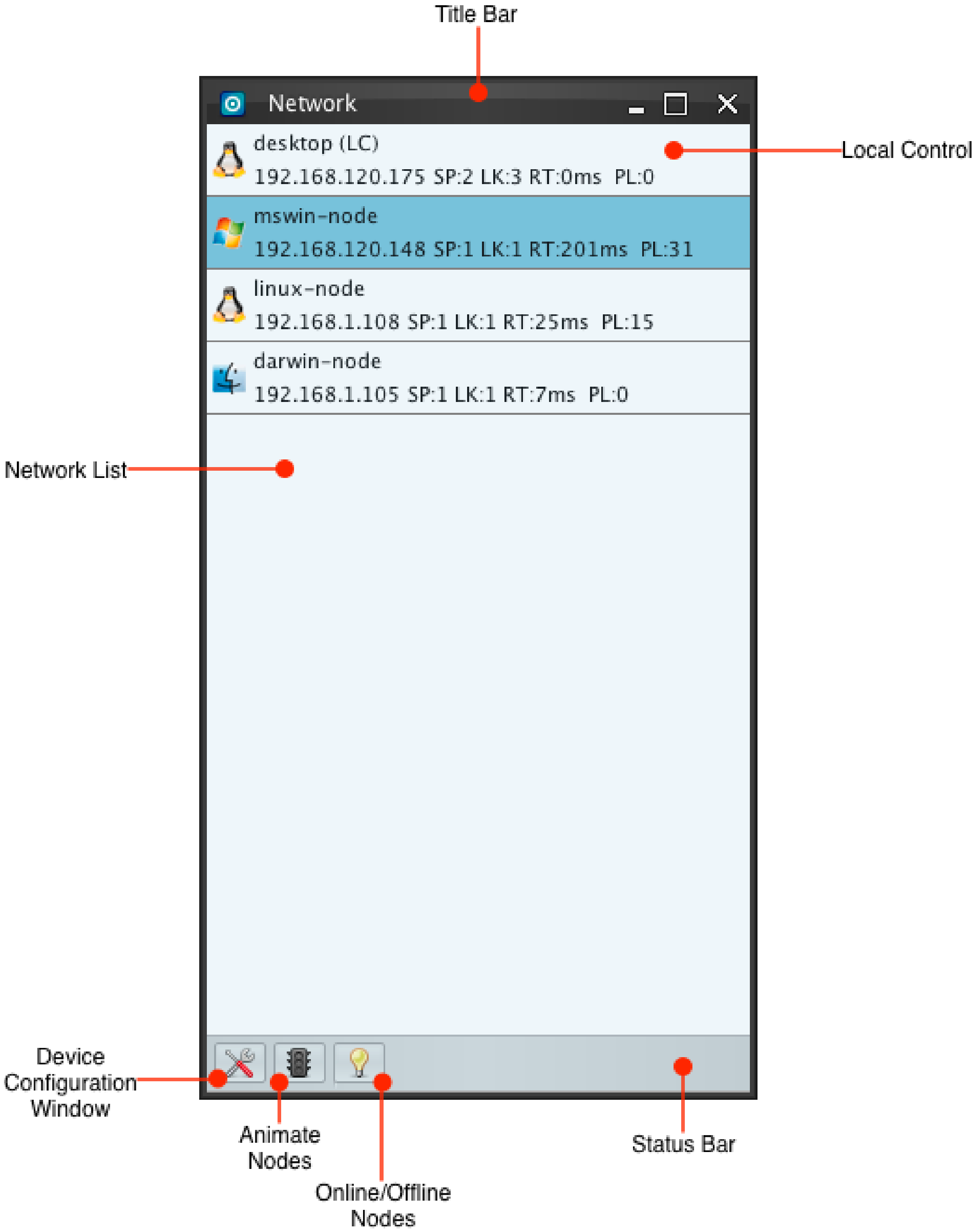
Displays the current authentication code read in from the .conf file. The authentication code can be copied to the clipboard by right clicking on the text field and selecting the Copy menu option.

5.2 Network Window

The Network Window is visible directly below the Dashboard when Horizon launches. The Network Window can be closed and reopened by clicking the “Network Window” button on the Dashboard. If the Network Window is already visible the window is brought to front.

The Network Window contains a list of all Controls and Nodes comprising the mesh network accessible from client’s Local Control and a Status Bar with the “Device Configuration” button.

The Controls and Nodes in the Network Window may be rearranged by dragging and dropping an item to another location. The Local Control, however, may not be moved and will be fixed to the top of the Network Window.



desktop (LC)
192.168.120.175 SP:2 LK:3 RT:0ms PL:0

mswin-node
192.168.120.148 SP:1 LK:1 RT:201ms PL:31

linux-node
192.168.1.108 SP:1 LK:1 RT:25ms PL:15

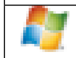
darwin-node
192.168.1.105 SP:1 LK:1 RT:7ms PL:0

Fig. 5.2: **Network Window**

The Network Window periodically updates status labels for each Control or Node device in the Network List; each status label includes the following.

- Operating System Icon: Icon indicates the operating system running the device. The following are the icons for various operating systems that may be displayed.

  Table 5.1: Operating System Icons Table

Icon	OS
	Linux
	Mac OS X
	Microsoft Windows
	Solaris
	FreeBSD
	NetBSD
	OpenBSD

- Device Name: The name provided in the Device Configuration Window's *Identity Panel* or the network-assigned hexadecimal stack address if no name is set.
- Control Status: Displays badge if the entry is a Control.

  Table 5.2: Control Status Table

Label	Meaning
(LC)	Indicates that the entry is the Local Control which is servicing the Horizon GUI's current session.
(C)	Indicates that the entry is a Control, but not the control servicing the Horizon GUI's current session.
(G)	Indicates that this node is the Transport module's *default gateway*.
Blank	Indicates that the entry is a Node.

- Device IP Address: The IP addresses associated with that device. The IP addresses are based on how the Local Control's device perceives each other device. In the case of network address translations, the IP address may be a private IP address for an internal LAN.
-

  Table 5.3: Shutdown Alarm State Table

Label	Meaning
(A)	Indicates the Node's shutdown alarm is set.
Blank	Indicates the Node's shutdown alarm is not set.

- SP #: The number of service ports currently supplied by the device. See the Device Configuration Window's *Service Ports Panel*.
- LK #: The number of links currently present on the device. See the Device Configuration Window's *Links Panel*.
- RT ##ms: The last reported round trip time in milliseconds. This value is calculated using running averages. The round trip time is updated periodically using ping messages or immediately when the user commands that particular device.
- PL##: Packet loss is a relative value used to indicate the number of lost packets, scaled using a multiplier, divided by sent packets. Packet loss combined with round trip time provides a clear indicator of connectivity issues with that particular device.

🛠 The "Device Configuration" button displays the *Device Configuration Window*, which allows manipulation of device settings including the service ports and links.

5.2.1 Link Quality Animation

🚦 Operators can toggle the "Animate Nodes" button to enable or disable a visual warning of devices with poor connectivity in the mesh. When the feature is enabled the button shows a lit traffic light 🚦. When a device's packet loss or round trip time exceeds warning or critical thresholds the entry begins to pulse yellow or red respectively. The threshold for packet loss and round trip time can be set in the Device Preferences panel of the *Device Window*.

5.2.2 Online/Offline Nodes

💡 Operators can toggle the "Online/Offline Nodes" button to hide/show nodes that have gone offline. Nodes that are no longer accessible will be marked as "(Offline)" and cannot be manipulated in the Device Configuration Window. The online state 💡 will hide all offline nodes. If a node comes back online with a matching Stack ID the node will no longer show up in the "Offline" state.

5.2.3 Manipulating Devices

Operators can a select a device in the Network List by left-clicking the mouse. Operators can right click to bring up a dynamically updated popup menu of actions pertaining to the selected device. The right click menu contains submenus for each module currently accessible for the selected device. Online and offline nodes have different menu options.

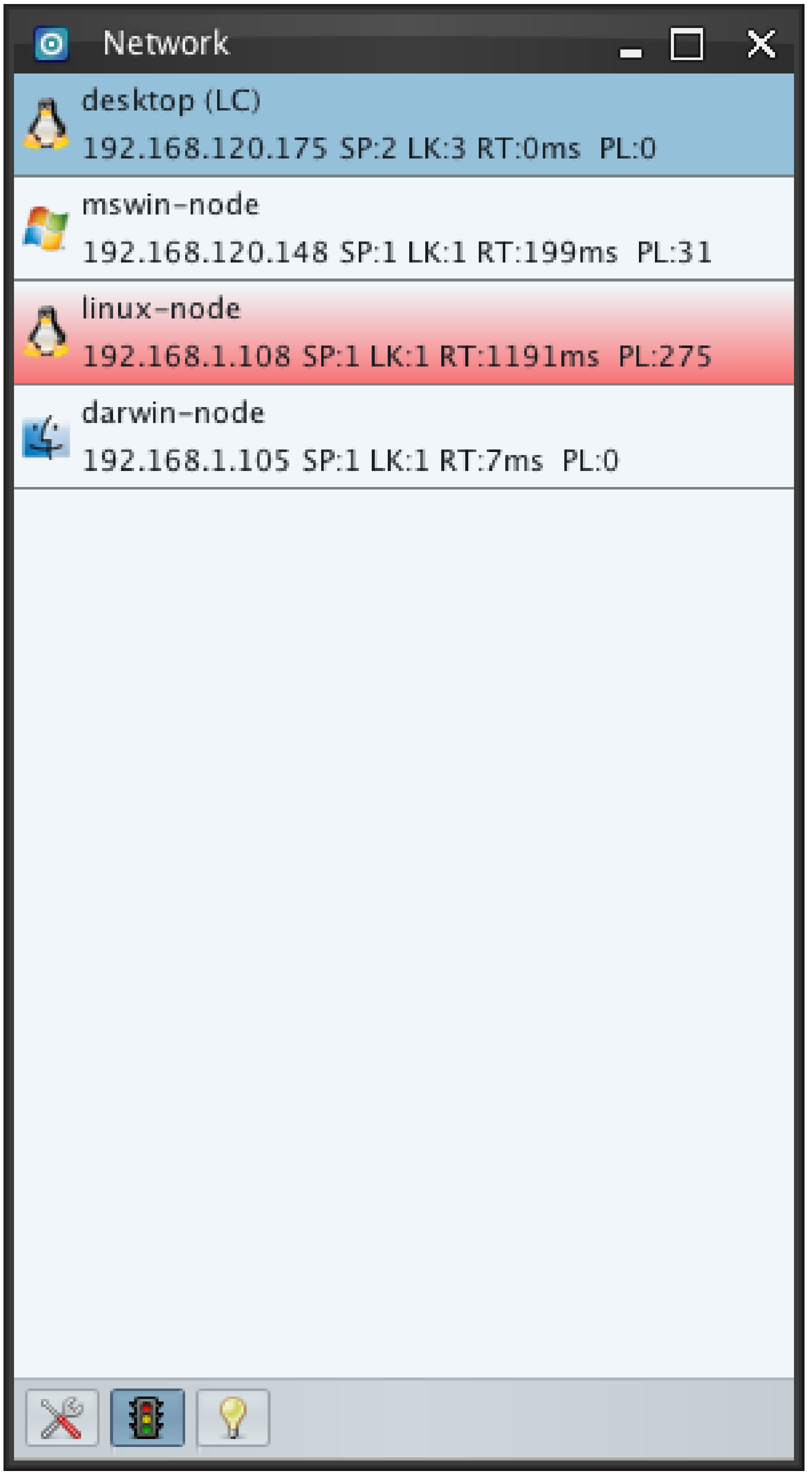
Online Nodes

The popup menu for online nodes contains submenus for each module currently accessible to the selected device. For a module's submenu to appear in the popup menu two conditions must be met.

1. The Local Control device must have the module activated.
2. The selected Control or Node device must have the module activated.

Modules can be activated through the Device Configuration Window's *Modules Panel*.

The Device module makes the following actions accessible though the Network Window popup menu.



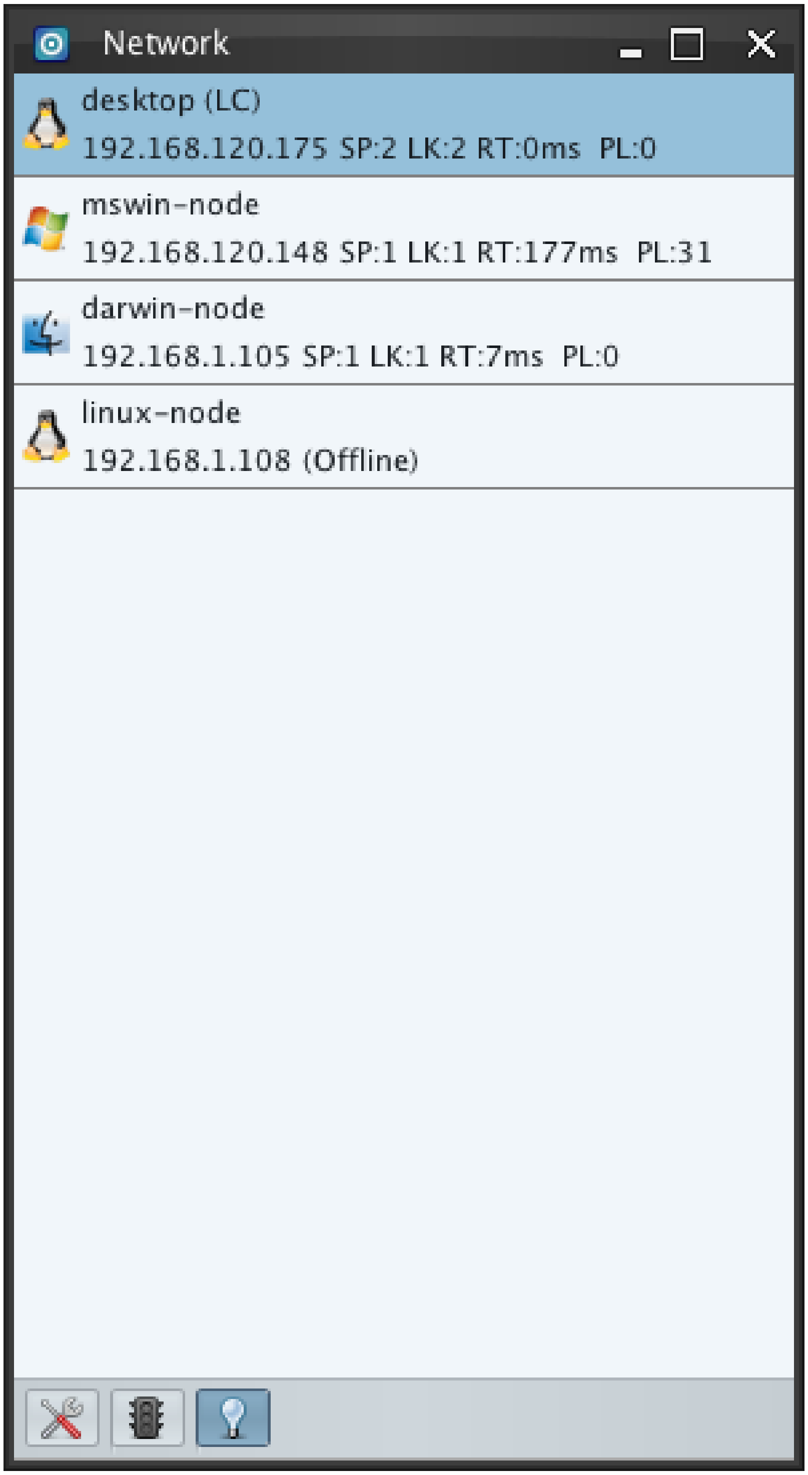
desktop (LC)
192.168.120.175 SP:2 LK:3 RT:0ms PL:0

mswin-node
192.168.120.148 SP:1 LK:1 RT:199ms PL:31

linux-node
192.168.1.108 SP:1 LK:1 RT:1191ms PL:275

darwin-node
192.168.1.105 SP:1 LK:1 RT:7ms PL:0

Fig. 5.3: **Network Window with a Critical Link Quality Animation**



desktop (LC)
192.168.120.175 SP:2 LK:2 RT:0ms PL:0

mswin-node
192.168.120.148 SP:1 LK:1 RT:177ms PL:31

darwin-node
192.168.1.105 SP:1 LK:1 RT:7ms PL:0

linux-node
192.168.1.108 (Offline)

Fig. 5.4: **Network Window with Offline Nodes Enabled**

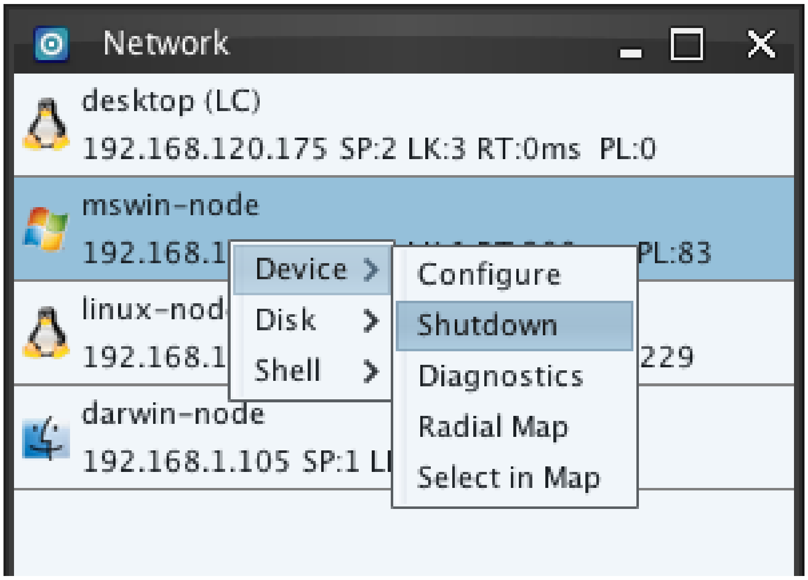
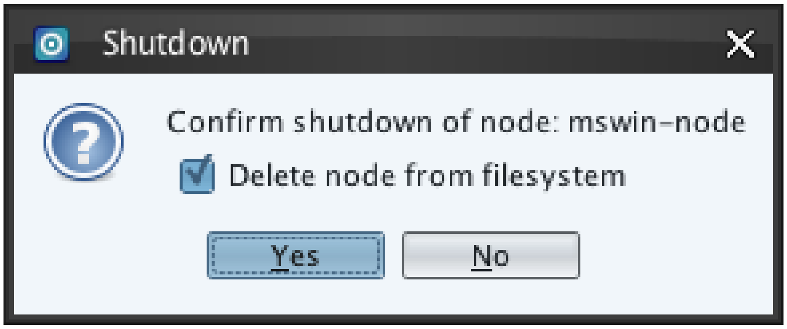


Fig. 5.5: **Right-click brings up Modules' Popup Menu**

1. Configure: This item displays the *Device Configuration Window*, which allows manipulation of device settings including the service ports and links. This is the same function as the 🛠 button.
2. Shutdown: This item prompts for conformation to shutdown the selected device with a dialog box. When shutting down Nodes, the Shutdown Dialog box contains a check box option "Delete Node from Filesystem" that instructs the device to delete its executable file from the system before exiting.

   The Horizon GUI's Local Control can be shutdown also by clicking the ⏻ button on Dashboard *Local Shutdown*.

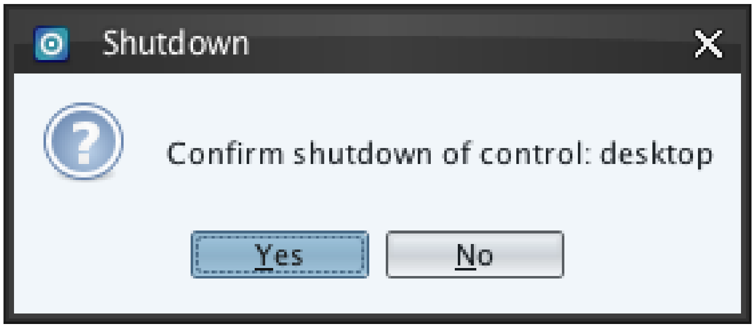


Confirm shutdown of node: mswin-node

☑ Delete node from filesystem

Fig. 5.6: **Device Shutdown Dialog Box for Nodes**

3. Diagnostics: This item brings up the *Diagnostics Window* for the currently selected device.
4. Radial Map: This item sets the selected device as the central device in a *Radial Map* view of the *Visual Map Window*.
5. Select in Map: This item opens the *Visual Map Window* and changes the current node selected in the Visual Map to match the selection in the Network List.



Confirm shutdown of control: desktop

Fig. 5.7: **Device Shutdown Dialog Box for Controls**

Offline Nodes

The Offline right click menu contains submenus for Device and Disk modules. The Device module makes the following actions accessible.

1. Info: This item displays the *Device Configuration Window*, similar to the online "Configure" option. The last known properties of the offline device are displayed for informative purposes.
2. Offline Map: This item opens a window with a screenshot of the Visual Map's (linear map) view right before the device was disconnected from the network. This option is only accessible for devices that disconnect with Horizon connected.
3. Remove: This item removes the currently selected offline device from the Network List.

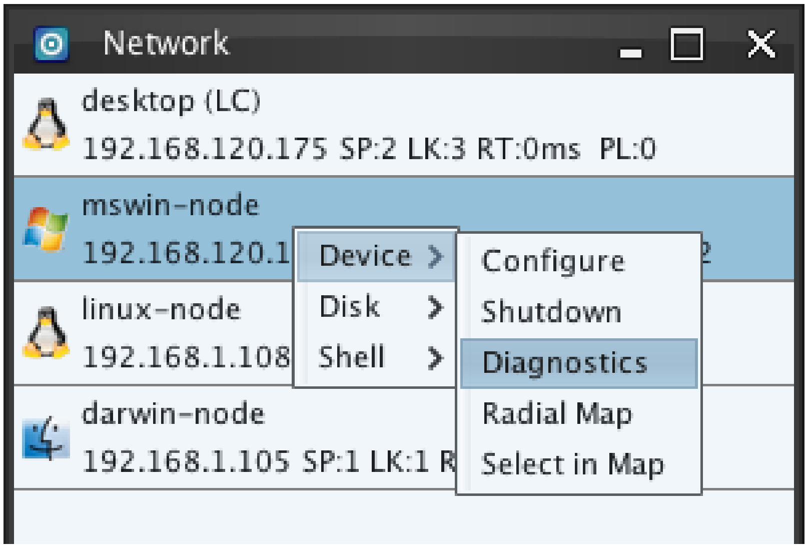
The offline right click menu's Disk submenu contains the an item for *Directory Listing* in offline browsing mode. Contents from previous online directory browsing or directory walk operations can be traversed.

5.3 Diagnostics Window

The Diagnostic Window provides detailed link information about the selected Node or Control and its connections within the mesh network. This window is accessed by right-clicking on a Node or Control in the Network Window and selecting the Diagnostic option in the Device Menu. A separate Diagnostic Window can be accessed for each device listed in the Network Window.

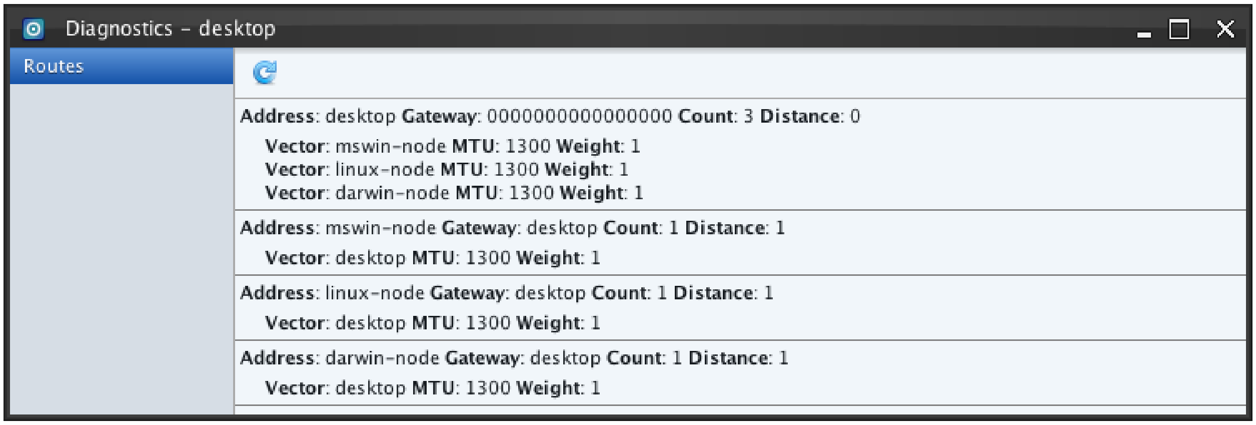
At the top of the Route List is the root Node or Control as denoted by the Gateway address 0000000000000000. Each entry below this represents another Node or Control that is connected, either directly or indirectly, to the selected Node or Control. The entries contain the following information:

- Address: Stack address of the device.
- Gateway: Stack address of the next Node or Control used to reach the root Node or Control.
- Count: Total number of direct links to other Nodes or Controls.
- Distance: Number of Nodes between itself and the root Node or Control.
- Link(s): Each link associated with this Node or Control is listed with the following information: Vector, MTU, and Weight.
  - Vector: Stack address of the device the link is connected to.
  - MTU: Maximum Transmission Unit of the link.
  - Weight: Priority rating.



```
desktop (LC)
192.168.120.175 SP:2 LK:3 RT:0ms  PL:0
mswin-node
192.168.120.1
linux-node
192.168.1.108
darwin-node
192.168.1.105 SP:1 LK:1 R
```

Fig. 5.8: **Diagnostic Window Menu Option**



```
Address: desktop Gateway: 0000000000000000 Count: 3 Distance: 0
   Vector: mswin-node MTU: 1300 Weight: 1
   Vector: linux-node MTU: 1300 Weight: 1
   Vector: darwin-node MTU: 1300 Weight: 1
Address: mswin-node Gateway: desktop Count: 1 Distance: 1
   Vector: desktop MTU: 1300 Weight: 1
Address: linux-node Gateway: desktop Count: 1 Distance: 1
   Vector: desktop MTU: 1300 Weight: 1
Address: darwin-node Gateway: desktop Count: 1 Distance: 1
   Vector: desktop MTU: 1300 Weight: 1
```

Fig. 5.9: **Device Diagnostic Window**

5.4 Device Window

The Device Window is a global window accessible from the Dashboard's Module combo box. This window provides an operator with the Device Preferences panel. An operator can open this window by selecting the Device option in the Module Combo Box:

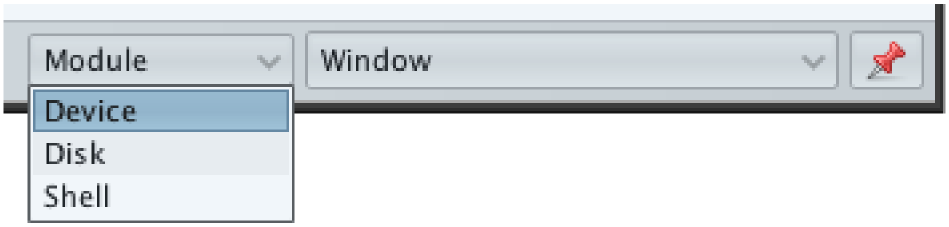
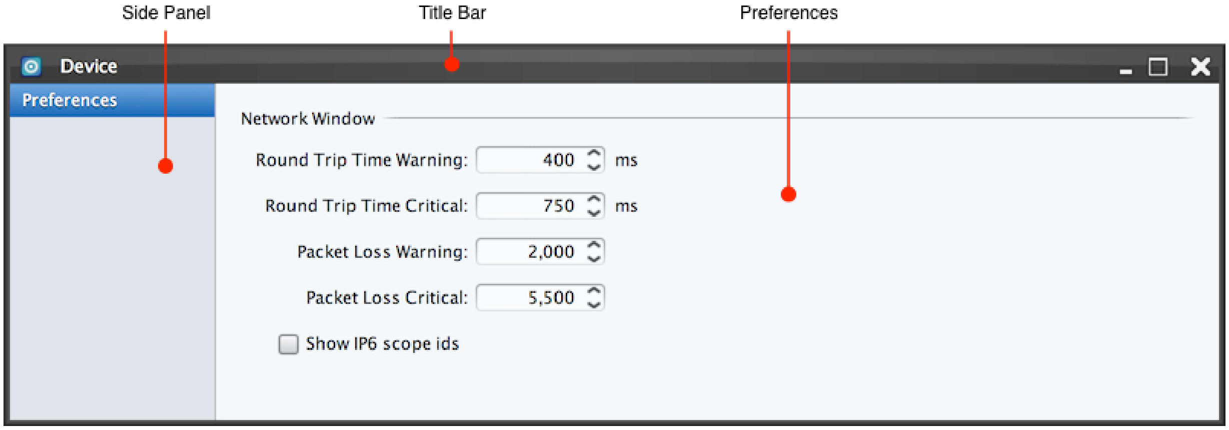


Fig. 5.10: **Dashboard Module Menu**

5.4.1 Preferences

Opening the Device Window and selecting the Preferences option in the side panel will display the Device Preferences panel:



Network Window

Round Trip Time Warning: 400 ms

Round Trip Time Critical: 750 ms

Packet Loss Warning: 2,000

Packet Loss Critical: 5,500

□ Show IP6 scope ids

Fig. 5.11: **Device Preferences Panel**

The Device Preferences panel consists of spin boxes to set the threshold boundaries for *Link Quality Animation*. The four fields contain default presets each time Horizon starts, but the values are not reset when the Dashboard connects and disconnects (*Connection Management*).

The show IPv6 scope ids checkbox toggles showing the network interface on link local IPv6 addresses, See *IP Version 6 Support*.

5.5 Device Configuration Window

The Device Configuration Window allows operators to manipulate the Network Window's selected device's properties. There is only one instance of the Device Configuration Window in the client; it always operates on the currently selected device in the Network List. The Device Configuration Window update accordingly when a new device is selected.

The “Device Configuration” button on the Network Window displays the “Device Configuration Window. If the Device Configuration Window is already visible it is brought to front.

Each button on the Device Configuration Window’s Tool Bar represents a group of device properties. Each group is displayed in a panel below the Tool Bar when their button is clicked. The panels include: Identity Panel, Service Ports Panel, Links Panel, Modules Panel, and Alarm Panel.

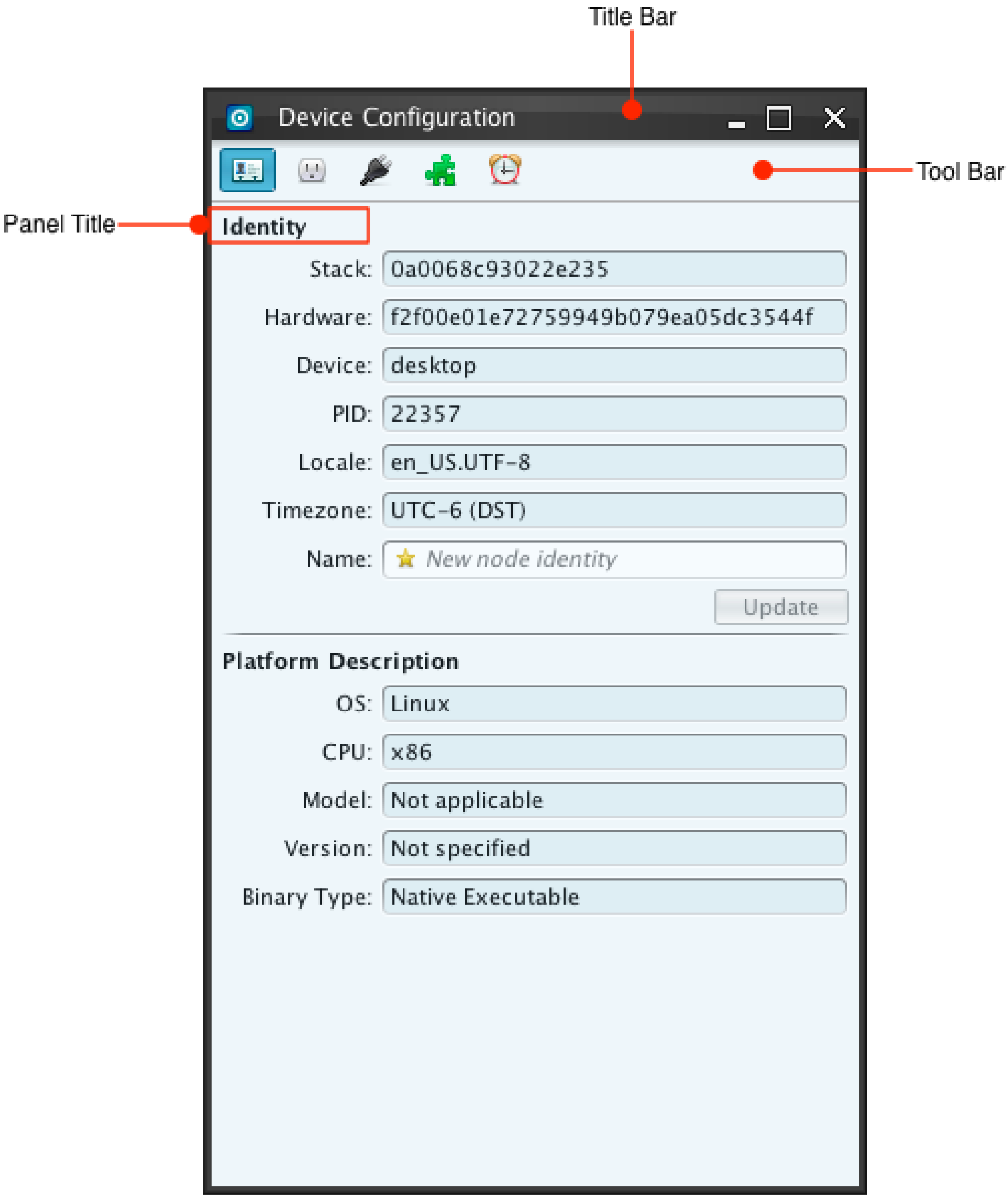
5.5.1 Identity Panel

The Identity panel displays the following identification information about the selected device.

- Stack: The stack address is a hexadecimal internal address that uniquely identifies devices throughout Fluxwire. The stack address is used as a label if “Device Name” is not set. The stack address changes each time a node runs.
- Hardware: The hardware value is a hexadecimal hash value that uniquely identifies a system that a node or control is running on. The hash value is created by collecting information about the hardware installed on a system.
- Device: The “Device Name” is an operator-defined label for a device.
- Hostname: The host system’s hostname and domain name. If the domain name is not available, only the hostname will appear.
- PID: The process identification number of the control or node on its host system.
- Locale: Language and character set, if set on host system.
- Timezone: Coordinated Universal Time with Daylight Savings, if set on host system.
- Name: This is an editable field the operator uses to change the “Device Name.”
- OS: The host system’s operating system.
- CPU: The host system’s CPU type.
- Platform Model: The host system’s model (i.e., linux:mipsle:paradyne), if any.
- Version: The host system’s version (i.e., linux:x86:mikrotik:5.x), if any.
- Binary Type: The type of executable used to deploy the node. This field displays one of the following values:
  - Native Executable: An executable program file type.
  - Dynamic Link Library: A Microsoft Windows dll file type. Some modules may have limited features on dll based Nodes. Refer to each module’s limitation’s section.

Setting the Device’s Name

Type a new name into the “Name” field and click the “Update” button. Once updated, the device name changes in all client components that display device information such as the Network Window, the Visual Map, and the “Device” field in the Identity Panel. To unset the Device Name empty the “Name” field (including spaces) and click the “Update” button. Once updated, client components display the hexadecimal stack address since no Device Name exists.



Identity

Stack: 0a0068c93022e235

Hardware: f2f00e01e72759949b079ea05dc3544f

Device: desktop

PID: 22357

Locale: en_US.UTF-8

Timezone: UTC-6 (DST)

Name: ☆ *New node identity*

Platform Description

OS: Linux

CPU: x86

Model: Not applicable

Version: Not specified

Binary Type: Native Executable

Fig. 5.12: **Device Configuration Window - Identity**

5.5.2 Service Ports Panel

Service ports are network sockets configured to receive incoming connections from other devices. Fluxwire supports one or more service ports of the following types: TCP and UDP. Service ports can use an optional application layer protocol. Service ports persist until an operator manually removes them or shuts down the device. As an operator adds service ports they are appended to the Service Ports List. The information displayed in this list includes:

- ID: Internal identifier for the service port.
- Address: Local IP address and the service port number.
- Type: TCP or UDP along with an optional protocol.

To remove a Service Port an operator must select a row in the Service Port List and choose the Remove option in the right-click menu.

Service Port Configure

To configure and add a new service port an operator must complete the following fields and click the 'Add' button.

- Type: TCP or UDP.
- IP Version: 4 for IP version 4, 6 for IP version 6, dual for dual-mode network sockets that support both IP version 4 or IP version 6 links.
- Port: Port number; 1-65535.
- Protocol: Application layer protocol; None, TCP/HTTP, TCP/HTTPS, UDP/RTP, or UDP/DNS.

Note: See the section on IPv6 support for details on how to use IP Version combo box. *IP Version 6 Support*.

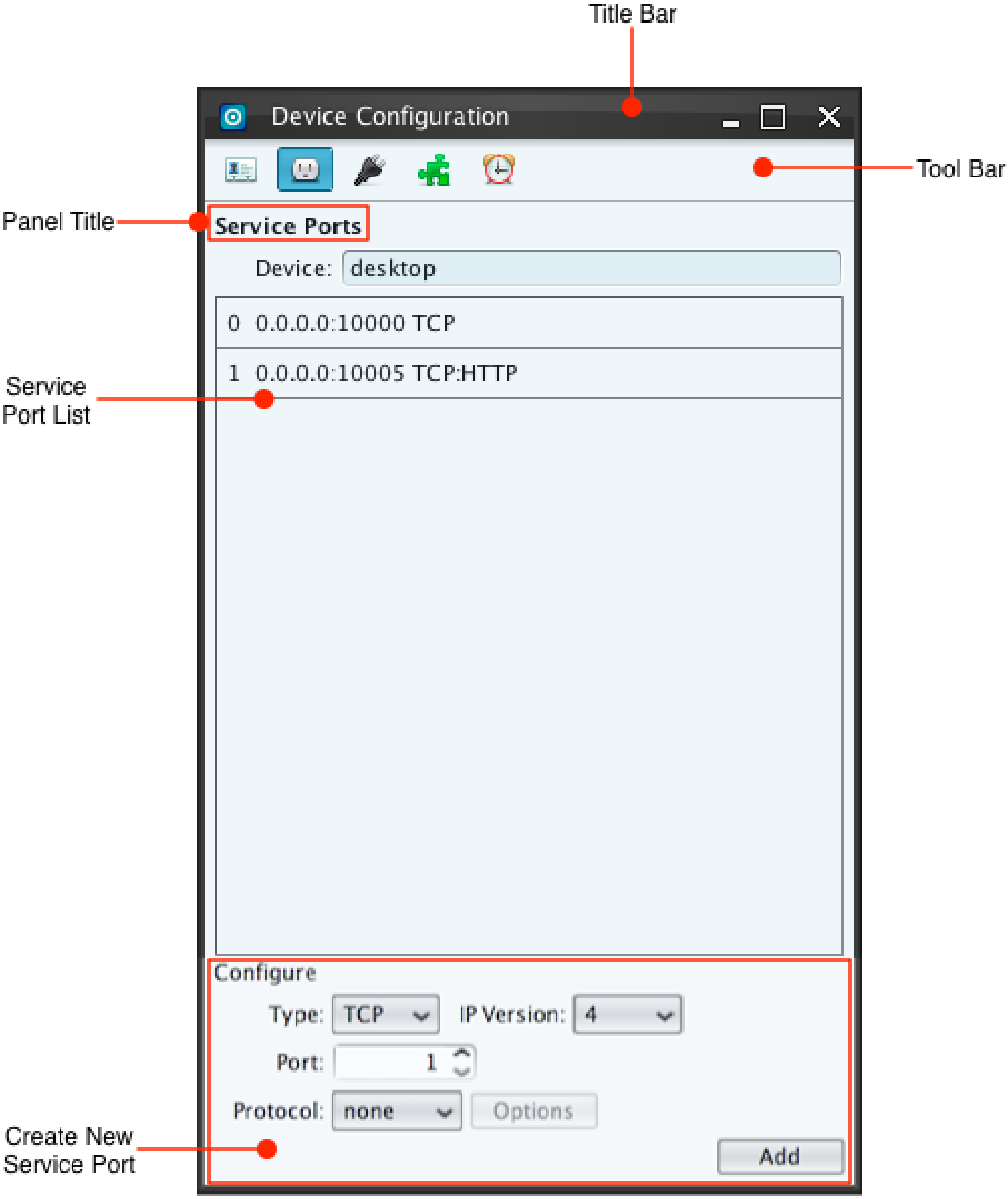
HTTP Service Port Options

- Server Response: Specifies how the HTTP server will encode data being sent back.
- Server Header Index: Index value of predefined HTTP response headers. 0-4 are valid in Microsoft Windows and 0-12 are valid on all Posix operating systems. See *HTTP(S) Server Response Options*.

HTTPS Service Port Options

- Server Response: Specifies how the HTTP server will encode data being sent back.
- Server Header Index: Index value of predefined HTTP response headers. 0-4 are valid in Microsoft Windows and 0-12 are valid on all Posix operating systems. See *HTTP(S) Server Response Options*.
- SSL Settings: For HTTPS only.
  - Private Key: Path to file containing RSA public certificate in PEM format.
  - Public Key: Path to file containing RSA private key in PEM format.
  - Crypto Library: Specifies the path to the crypto library from Openssl, called libcrypto, on the target system (Available only on Posix systems using the *ssl plugin*).

Note: HTTPS requires both http and ssl modules to be loaded. See *Modules Panel*.



Device: desktop

0 0.0.0.0:10000 TCP

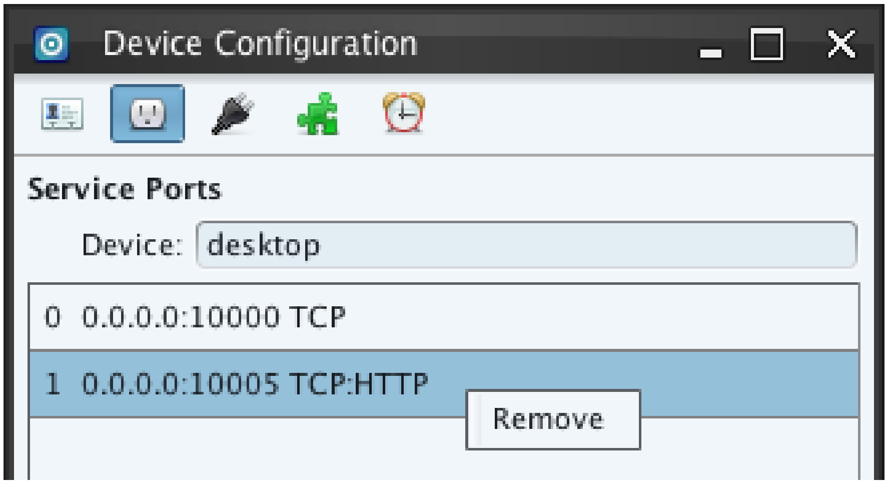
1 0.0.0.0:10005 TCP:HTTP

Type: TCP IP Version: 4

Port: 1

Protocol: none

Fig. 5.13: **Device Configuration Window - Service Ports**



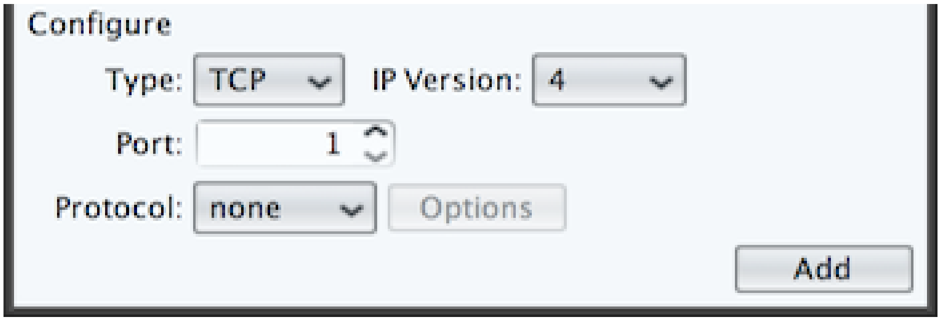
Service Ports

Device: desktop

0 0.0.0.0:10000 TCP

1 0.0.0.0:10005 TCP:HTTP

Fig. 5.14: **Device Configuration Window - Remove Service Port Popup Menu**



Configure

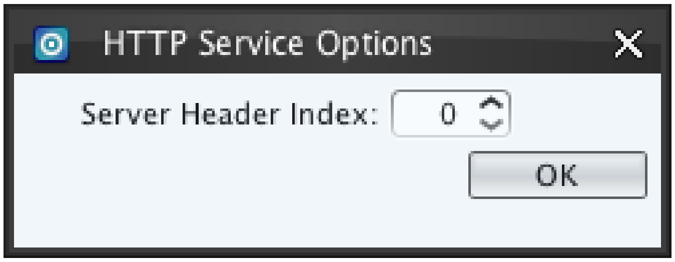
Type: TCP

IP Version: 4

Port: 1

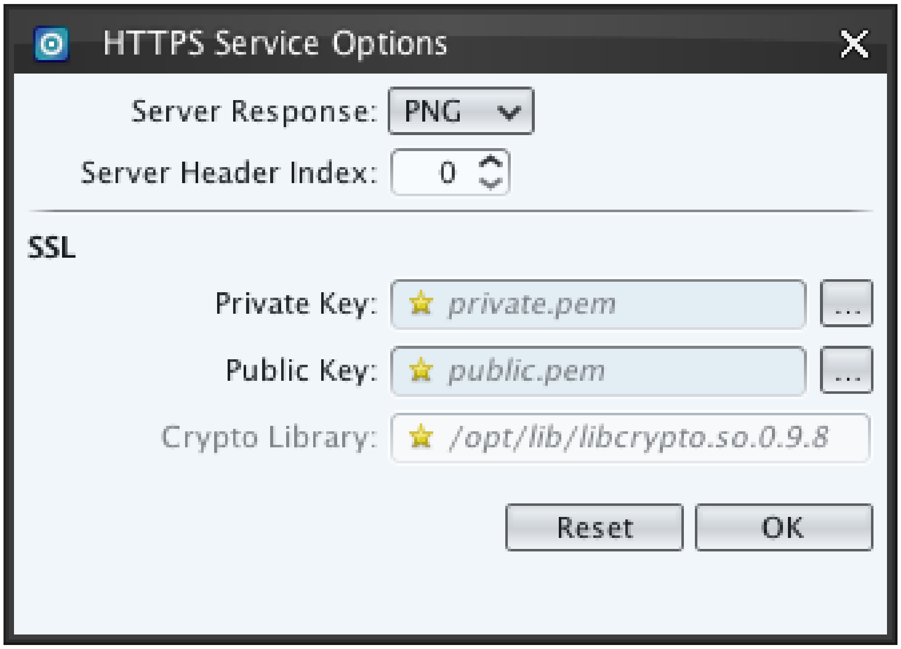
Protocol: none

Fig. 5.15: **Device Configuration Window - Service Port Configure**



Server Header Index: 0

Fig. 5.16: **Device Configuration Window - HTTP Service Port Options**



Server Response: PNG

Server Header Index: 0

SSL

Private Key: private.pem

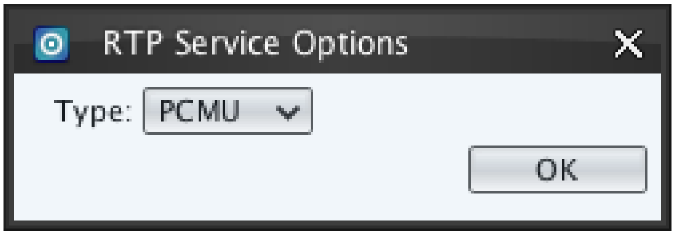
Public Key: public.pem

Crypto Library: /opt/lib/libcrypto.so.0.9.8

Fig. 5.17: **Device Configuration Window - HTTPS Service Port Options**

RTP Service Port Options

- Type: Specifies the encoding type to set in the RTP packet header (PCMU or PCMA).



Type: PCMU

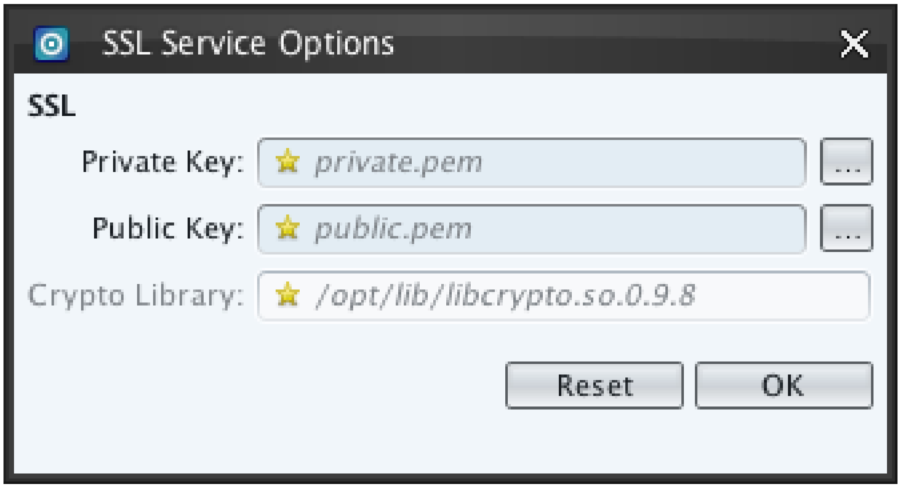
Fig. 5.18: **Device Configuration Window - RTP Service Port Options**

SSL Service Port Options

- Private Key: See SSL settings in *HTTPS Service Port Options*.
- Public Key: See SSL settings in *HTTPS Service Port Options*.
- Crypto Library: See SSL settings in *HTTPS Service Port Options*.

5.5.3 Links Panel

Links are active connections between a local and remote device. Fluxwire supports one or more links of the following types: TCP and UDP along with an optionally specified application layer protocol. UDP links provide an additional feature not available for the TCP links where a UDP link can be instantiated from an active UDP service port or from the local port of another active UDP link. TCP links will always be instantiated using a random local port. This unique



SSL Service Options

SSL

Private Key: private.pem

Public Key: public.pem

Crypto Library: /opt/lib/libcrypto.so.0.9.8

Fig. 5.19: **Device Configuration Window - SSL Service Port Options**

feature facilitates the reuse of UDP ports for both inbound and outbound links. New links are created by filling out the fields below the Links List. The Links List displays the following information:

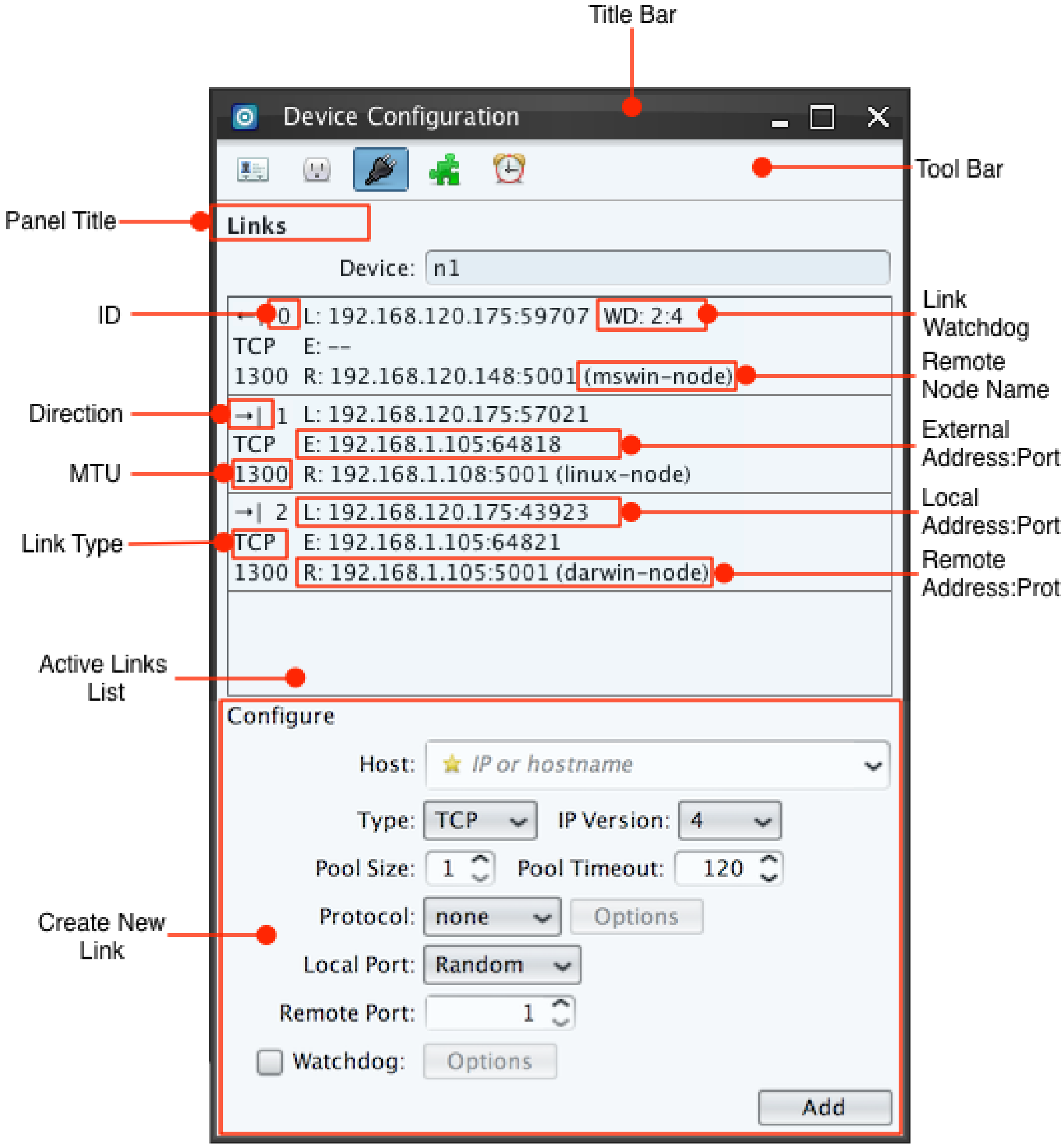
- ID: Internal identifier for the link.
- Direction: <-| for links originating from the device. ->| for links accepted by the device.
- Type: TCP or UDP and protocols optional protocols such as HTTP.
- Local Address: (IP:Port) Local IP address and the port for the link.
- External Address: (IP:Port) The public facing IP address when the node is on an internal LAN with Network Address Translation (NAT). This field displays "–" for links between two nodes in the same network address space or when both nodes have public facing IP addresses.
- MTU: Maximum Transmission Unit of the device.
- Remote Address: (IP:Port) Remote IP address and the port for the link.
- Remote Node Name: Remote device's name on the opposite side of the link.
- Watchdog Status: Status of the link watchdog in delay:retries format. Delay is in minutes. If no watchdog set or this the accepting side of a link, watchdog status will not appear.

To remove a Link an operator must select a row in the Links List and choose the Remove option in the right-click menu.

Link Configure

To configure and add a new link an operator must complete the following fields and click the 'Add' button.

- Host: The IP address or domain name for the remote host. The domain name will be resolved locally before continuing.
- Link Type: TCP or UDP.
- IP Version: 4 for IP version 4. 6 for IP version 6. 4,6 for Host resolution preference of IPv4 over IPv6. 6,4 for Host resolution preference of IPv6 over IPv4.
- Pool Size: Applies only to TCP. The pool is the number of underlying TCP connections representing a Fluxwire Link. Values can range from 1 (no pooling) to 8 pooled TCP connections. For HTTP(s) protocols the pool



Device Configuration

Links

Device: n1

```
←| 0  L: 192.168.120.175:59707 WD: 2:4
TCP   E: --
1300  R: 192.168.120.148:5001 (mswin-node)
→| 1  L: 192.168.120.175:57021
TCP   E: 192.168.1.105:64818
1300  R: 192.168.1.108:5001 (linux-node)
→| 2  L: 192.168.120.175:43923
TCP   E: 192.168.1.105:64821
1300  R: 192.168.1.105:5001 (darwin-node)
```

Configure

Host: IP or hostname

Type: TCP IP Version: 4

Pool Size: 1 Pool Timeout: 120

Protocol: none Options

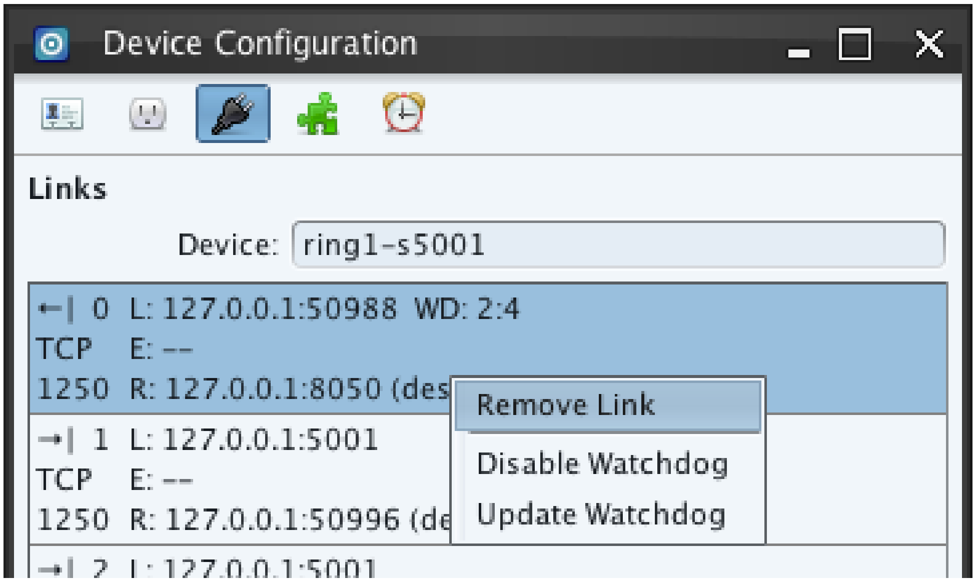
Local Port: Random

Remote Port: 1

□ Watchdog: Options

Add

Fig. 5.20: **Device Configuration Window - Links**



Links

Device: ring1-s5001

```
←| 0   L: 127.0.0.1:50988  WD: 2:4
TCP    E: --
1250   R: 127.0.0.1:8050 (des
→| 1   L: 127.0.0.1:5001
TCP    E: --
1250   R: 127.0.0.1:50996 (de
→| 2   L: 127.0.0.1:5001
```

Fig. 5.21: **Device Configuration Window - Remove Link Popup Menu**

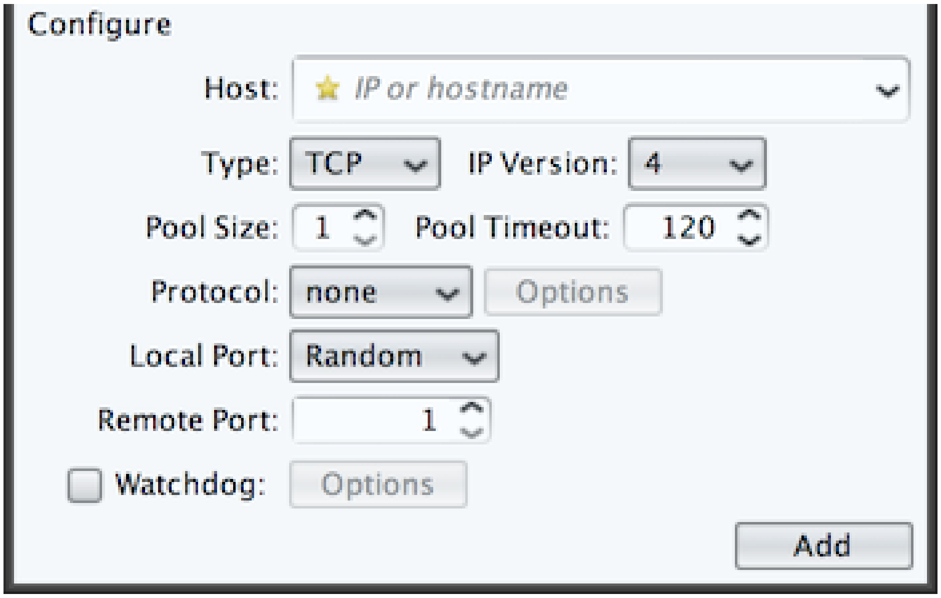
represents the number of http requests backing the link; more connections can increase HTTP(s) throughput. For all other TCP links, TCP connections are staggered when created and destroyed. Therefore, the pool size represents the maximum at any given time, but the actual number will likely be lower at any given time.

- Pool Timeout: The absolute minimum time in seconds a pooled TCP link will last before disconnecting. The minimum value is 60 seconds. Links will randomize their connection length by up to an additional 60 seconds. Therefore the minimum duration can be is between 1 to 2 minutes. This is not used for any UDP links.
- Protocol: Application layer protocol, None or HTTP. When a protocol is selected, protocol specific options are accessible through the "Options" button.
- Local Port: "Random" selects a random unused port number. "Custom" allows a user specified port number. When "Custom" is selected a spin box appears for users to enter the port number. UDP service port numbers can also be used for UDP links. UDP service port numbers are preloaded in the combo box.
- Remote Port: The remote device's service port to connect to.
- Watchdog: Checking watchdog enables a link watchdog. Use the Options button to the right set the settings for the watchdog. The options dialog retains their previous values for reuse.

Note: See the section on IPv6 support for details on how to use IP Version combo box. *IP Version 6 Support*.

HTTP and HTTPS Link Options

- Host: URI (Uniform Resource Identifier) that will be written to the host, HTTP Field in an HTTP request.
- Port: An optional HTTP URI port number. This can be useful with proxy mode to force proxies to go to a non-standard http service port.
- URI Depth: The number of words in a directory pattern for GET and POST requests. E.G. one/two/three/script.php
- Proxy Settings: Check proxy checkbox to enable proxy support. * Proxy Credentials: User and password credential pairs to be used for basic proxy authentication. The format accepted is *user1:pass1,user2:pass2*



Configure

Host: IP or hostname

Type: TCP

IP Version: 4

Pool Size: 1

Pool Timeout: 120

Protocol: none

Local Port: Random

Remote Port: 1

□ Watchdog:

Fig. 5.22: **Device Configuration Window - Link Configure**

- SSL Settings: For HTTPS only.
  - Crypto Library: Specifies the path to the crypto library from Openssl, called libcrypto, on the target system (Available only on Posix systems using the *ssl plugin*).

RTP Link Options

- Type: Specifies the encoding type to set in the RTP packet header (PCMU or PCMA).

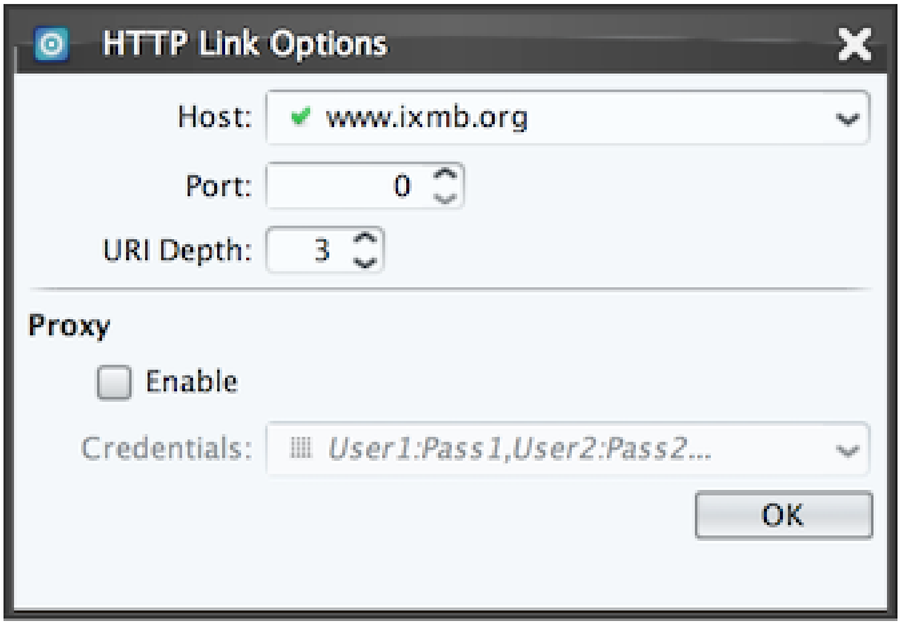
DNS Link Options

- Domain: Domain name

SSL Link Options

Nodes can leverage three different libraries implementing the SSL cryptography functions; native Windows library for MS Window systems, native OpenSSL library on Posix based systems and PolarSSL plugin for Posix based systems lacking a native library. The PolarSSL plugin is either bundled into the node using the packer or dynamically loaded at link configuration.

- Crypto Library: See SSL settings in *HTTPS Service Port Options*.
- **Mode: Specifies the mode.**
  - Full: Handles an initial SSL handshake and tunnels the data using SSL.
  - Partial: Handles an initial SSL handshake but disables SSL for data relying only on the internal protocol for data encryption.


HTTP Link Options

Host: ✓ www.ixmb.org

Port: 0

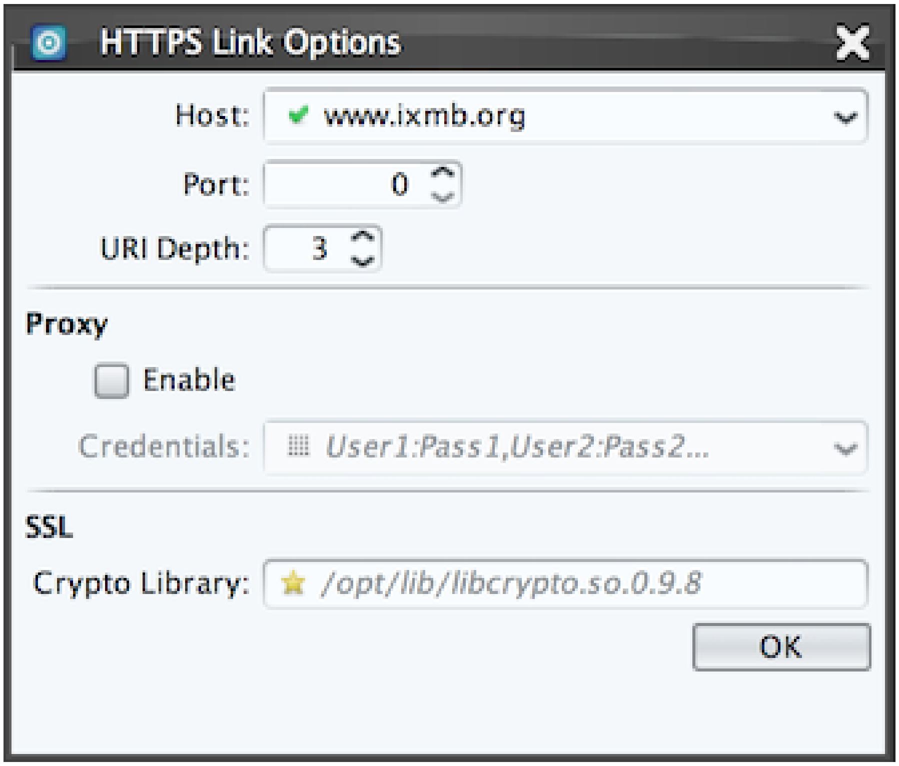
URI Depth: 3

Proxy

□ Enable

Credentials: User1:Pass1,User2:Pass2...

Fig. 5.23: **Device Configuration Window - HTTP Link Options**


HTTPS Link Options

Host: ✓ www.ixmb.org

Port: 0

URI Depth: 3

Proxy

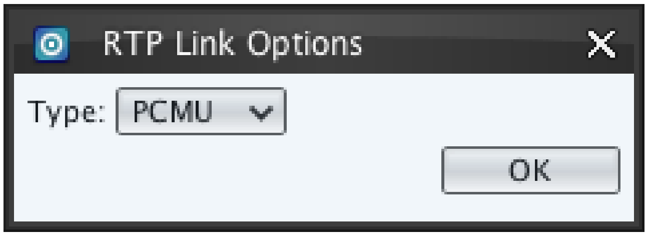
□ Enable

Credentials: User1:Pass1,User2:Pass2...

SSL

Crypto Library: /opt/lib/libcrypto.so.0.9.8

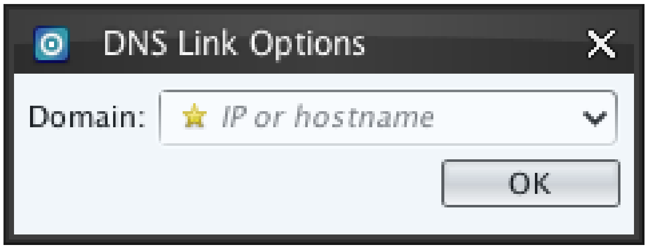
Fig. 5.24: **Device Configuration Window - HTTPS Link Options**



RTP Link Options

Type: PCMU

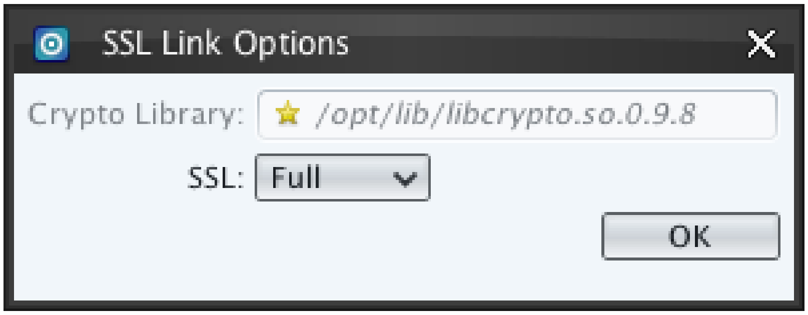
Fig. 5.25: **Device Configuration Window - RTP Link Options**



DNS Link Options

Domain: IP or hostname

Fig. 5.26: **Device Configuration Window - DNS Link Options**



SSL Link Options

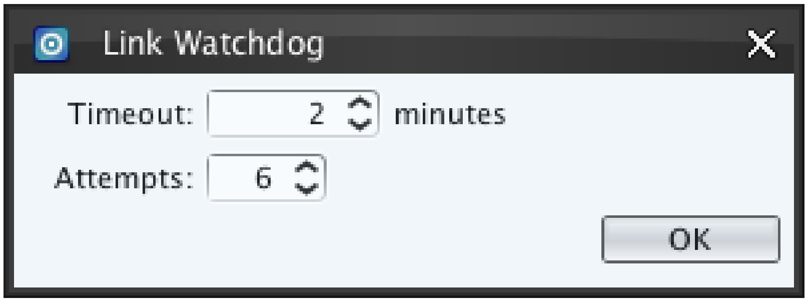
Crypto Library: /opt/lib/libcrypto.so.0.9.8

SSL: Full

Fig. 5.27: **Device Configuration Window - SSL Link Options**

Link Watchdog

Link watchdogs will attempt to reestablish existing links if the link disconnects or times out. Link watchdogs can only be configured on the originating side **|link-orig|** of the link. Watchdogs can be set during creation (*Link Configure*). Link watchdogs require two values to configure, a delay timeout in minutes to wait before attempting to reestablish the link and the number of attempts to retry before giving up. Devices will wait the specified delay timeout between each attempt. The link watchdog settings can be modified or disabled on the originating side of existing links by right clicking a link in the Active Link List.



Timeout: 2 minutes

Attempts: 6

Fig. 5.28: **Device Configuration Window - Link Watchdog Options**

> **Warning:** Watchdog timeouts do not start until the Operating System times out the link. This can vary especially with TCP settings. The Alarm timeout (*Alarm Panel*) should be set accordingly to allow any link watchdogs enough time to reestablish connectivity.

5.5.4 Modules Panel

The Module Panel allows individual modules to be activated and deactivated for the currently selected device in the Network Window. The panel's Module List shows all possible modules and marks each loaded module as "(active)".

Modules can be loaded and unloaded by selecting and right-clicking a module in the Module List. A pop-up menu appears with two actions, "Load" and "Unload." If the module is currently active only "Unload" is accessible and vice versa.

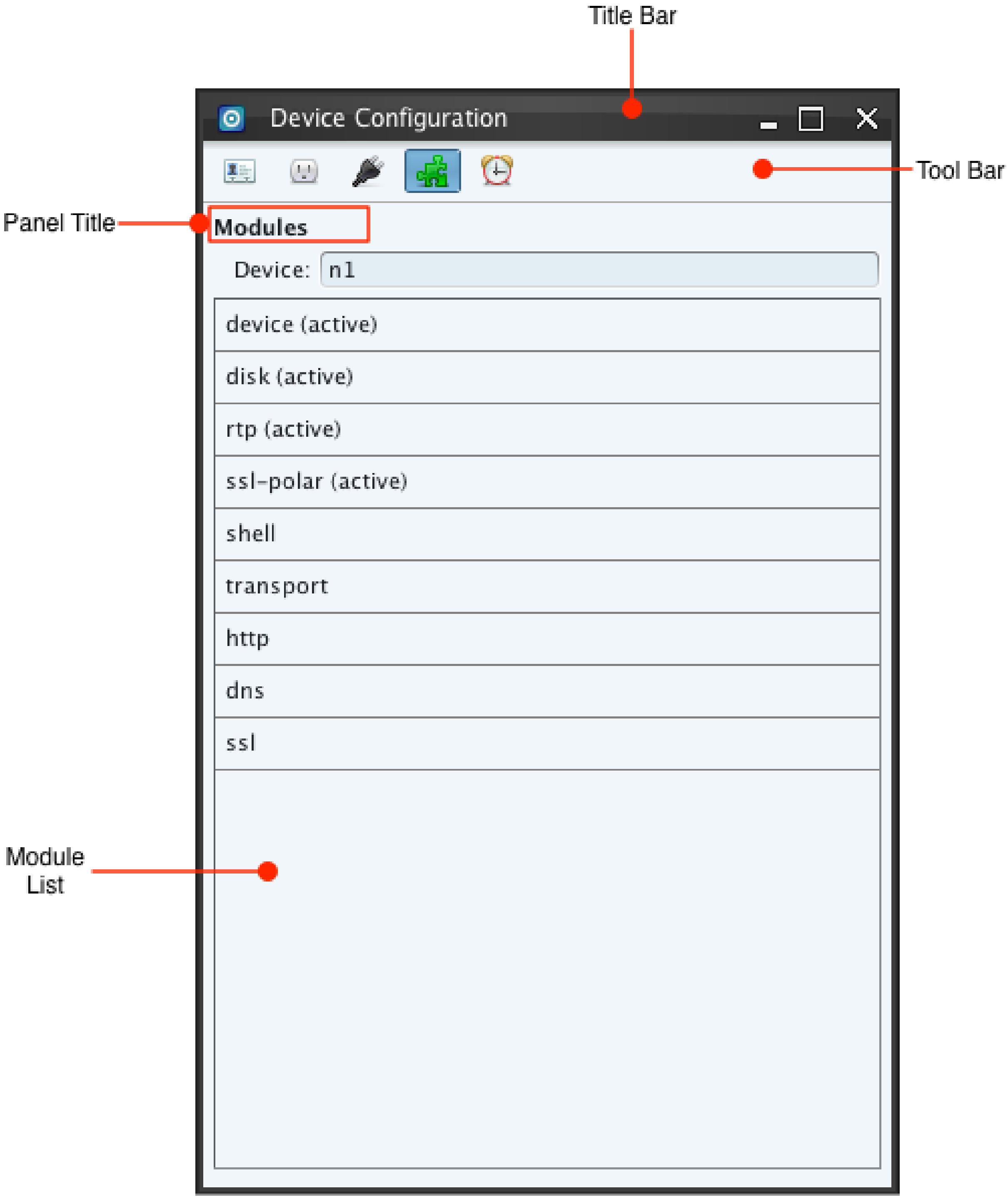
Blacklist

Modules that are included in the Blacklist (BL) parameter when building a node using *Packer (flx-packer)* cannot be loaded on a node. This prevents modules from being loaded both automatically and manually. If an attempt is made to load a blacklisted module, it will be blocked and a log message will appear in the Logs window.

5.5.5 Alarm Panel

The Alarm Panel specifies shutdown alarm parameters for the currently selected Node device in the Network Window; alarms cannot be set for Controls. Nodes periodically broadcasts a ping message to all Controls and waits for a response. If a response is received within the timeout period, the alarm resets, otherwise the alarm triggers a shutdown. The Alarm Panel displays the current settings in the Alarm Status section, user interface components to configure the alarm, and two buttons to start and stop the alarm.

Each item in the Alarm Panel is broken down in more detail below.



Device Configuration

Modules

Device: n1

device (active)

disk (active)

rtp (active)

ssl-polar (active)

shell

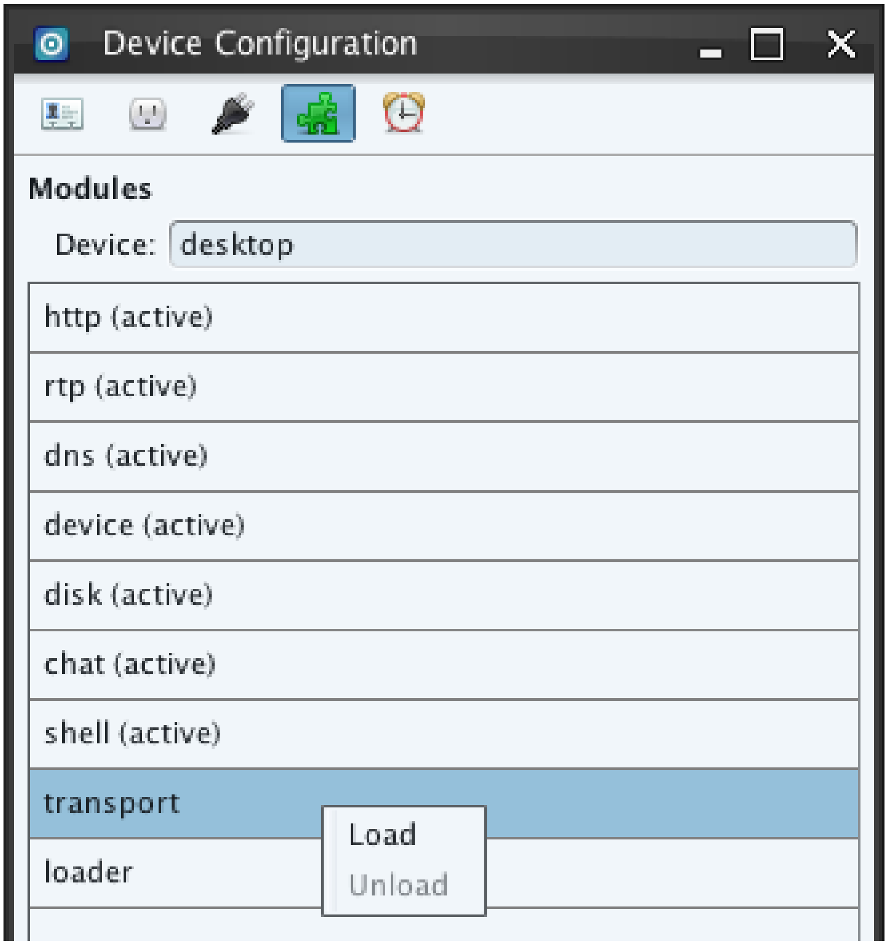
transport

http

dns

ssl

Fig. 5.29: **Device Configuration Window - Modules**

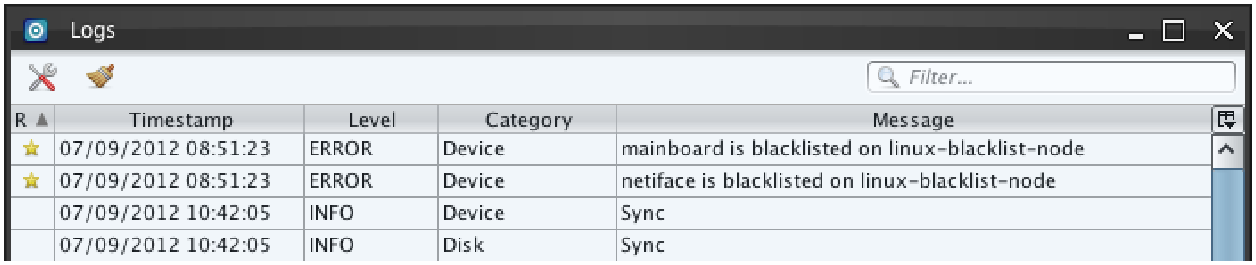


Modules

Device: desktop

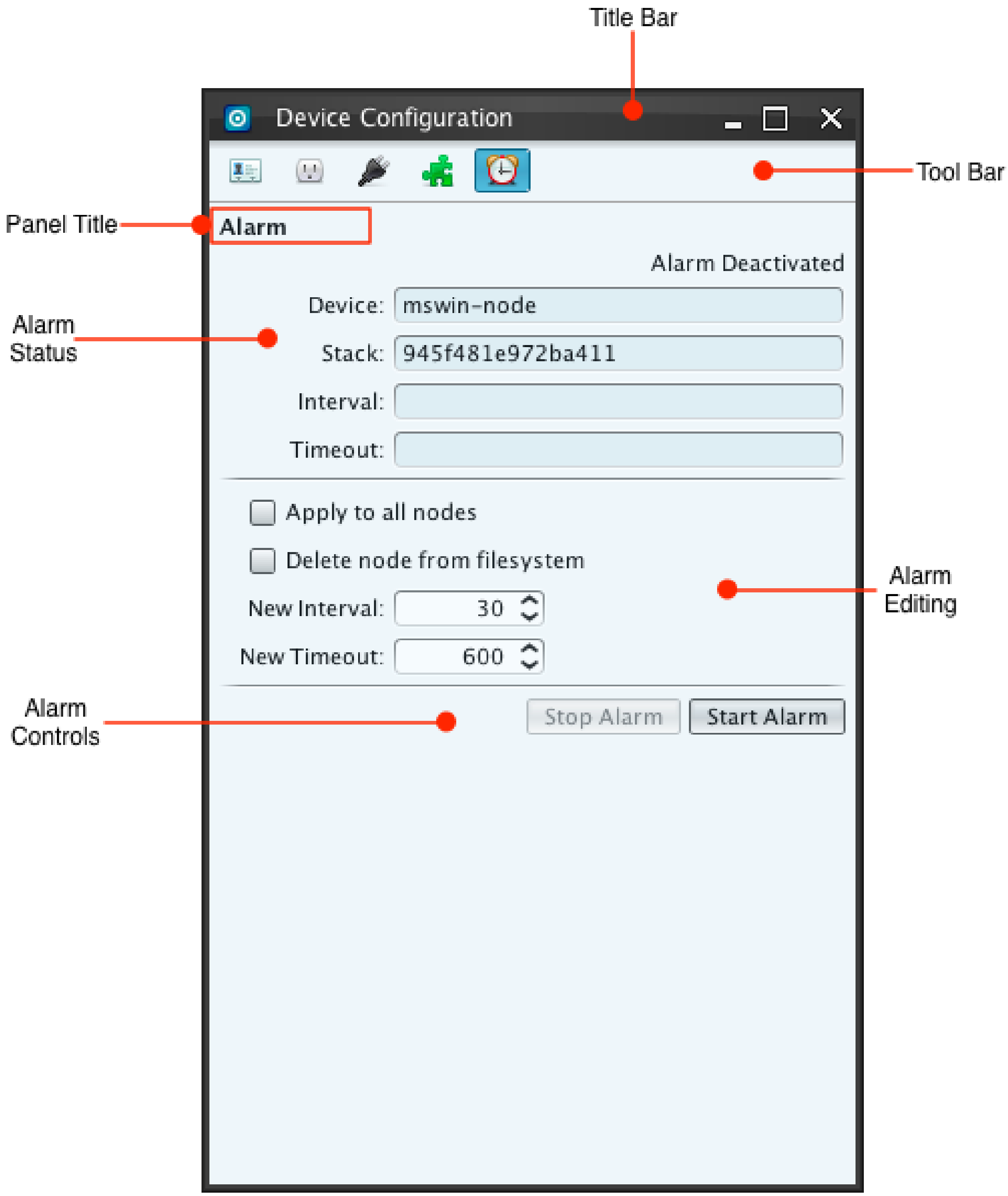
- http (active)
- rtp (active)
- dns (active)
- device (active)
- disk (active)
- chat (active)
- shell (active)
- transport
- loader

Fig. 5.30: **Device Configuration Window - Modules Panel's Popup Menu**



R ▲	Timestamp	Level	Category	Message
★	07/09/2012 08:51:23	ERROR	Device	mainboard is blacklisted on linux-blacklist-node
★	07/09/2012 08:51:23	ERROR	Device	netiface is blacklisted on linux-blacklist-node
	07/09/2012 10:42:05	INFO	Device	Sync
	07/09/2012 10:42:05	INFO	Disk	Sync

Fig. 5.31: **Blacklisted Module Load Error**



Device Configuration

Alarm

Alarm Deactivated

Device: mswin-node

Stack: 945f481e972ba411

Interval:

Timeout:

□ Apply to all nodes

□ Delete node from filesystem

New Interval: 30

New Timeout: 600

Fig. 5.32: **Device Configuration Window - Alarm**

- Alarm Active Notification: A status display that informs if the alarm is active or inactive.
- Device: The Device Name is a operator-defined label for a device specified in the Identity Panel.
- Stack: The stack address is a hexadecimal internal address that uniquely identifies devices in Fluxwire. The stack address is used as a label if the Device Name is not set.
- Interval: Displays the current the interval between broadcasting ping messages to Controls if an alarm is set.
- Timeout: Displays the current timeout in seconds that the Node waits for response from a Control before the alarm triggers.
- Apply to All Nodes: This check box is for configuring alarms. If this box is checked when the "Start Alarm" button is clicked, alarms are created for all Nodes in the network.
- Delete Node from Filesystem: Checking this box when the "Start Alarm" button is clicked instructs the alarm to delete the Node device's executable file(s) in addition to shutting down when the alarm is triggered.
- New Interval: Specifies the interval in seconds between broadcasting ping messages to all controls.
- New Timeout: Specifies the timeout in seconds that the Node waits for a response from a Control before the alarm triggers the Node to shutdown.
- Start Alarm: This button sets the alarm. If an alarm is already set, this button is disabled and "Stop Alarm" must be clicked first to cancel the existing alarm.
- Stop Alarm: The button stops alarms for the currently selected Node when "Apply to All Nodes is not checked; when "Apply to All Nodes' is checked all Nodes' alarms are stopped.

Note: "Start Alarm" and "Stop Alarm" buttons are always accessible when "Apply to All Nodes" box is checked. Nodes that already have alarms set cancel their old alarm and replace it when "Apply to All Nodes" is checked. Clicking "Stop Alarm" when "Apply to All Nodes" is checked cancels all Nodes' alarms.

Warning: Once an alarm timer has been set with the "Delete Node from Filesystem," the delete option cannot be undone. Even if the alarm timer is later stopped, the node's executable will be removed from the filesystem. The same conditions apply to timers configured at startup using the packer tools *Timeout (-i)*. The Alarm timeout should be set accordingly to allow *Link Watchdog* enough time to reestablish connectivity when set.

5.6 Visual Map Window

The Visual Map Window provides a real-time display of all active devices on the mesh network. The default "Linear Map" mode displays devices and their link topology. Multiple links can exist between any two devices, so the Visual Map is capable of grouping multiple links into one edge or rendering all links as parallel edges. When links are grouped together the labels display a count of how many link types exist for each time.

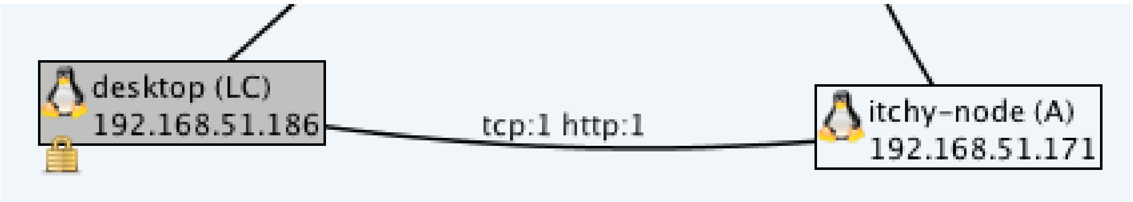


Fig. 5.33: **Edge Representing 2 links: 1 TCP and 1 TCP using Http protocol**

Every device in the Network Window displays several configurable attributes described in *Customizing Nodes*. The Local Control is always highlighted in a light gray color. The currently selected device is drawn with a thick blue border. Various configuration options are available for displaying attributes of devices and links. The Visual Map attempts to pack the device label information as efficiently as possible to save space. Therefore, status badges like packet loss and alarms may display in different orders for each node.

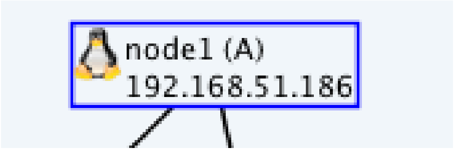


Fig. 5.34: **Device Selected with Blue Border and Alarm Set**

5.6.1 Showing the Visual Map

Pressing the "Visual Map" button on the Dashboard displays the Visual Map Window. If the Visual Map is already visible it is brought to the front.

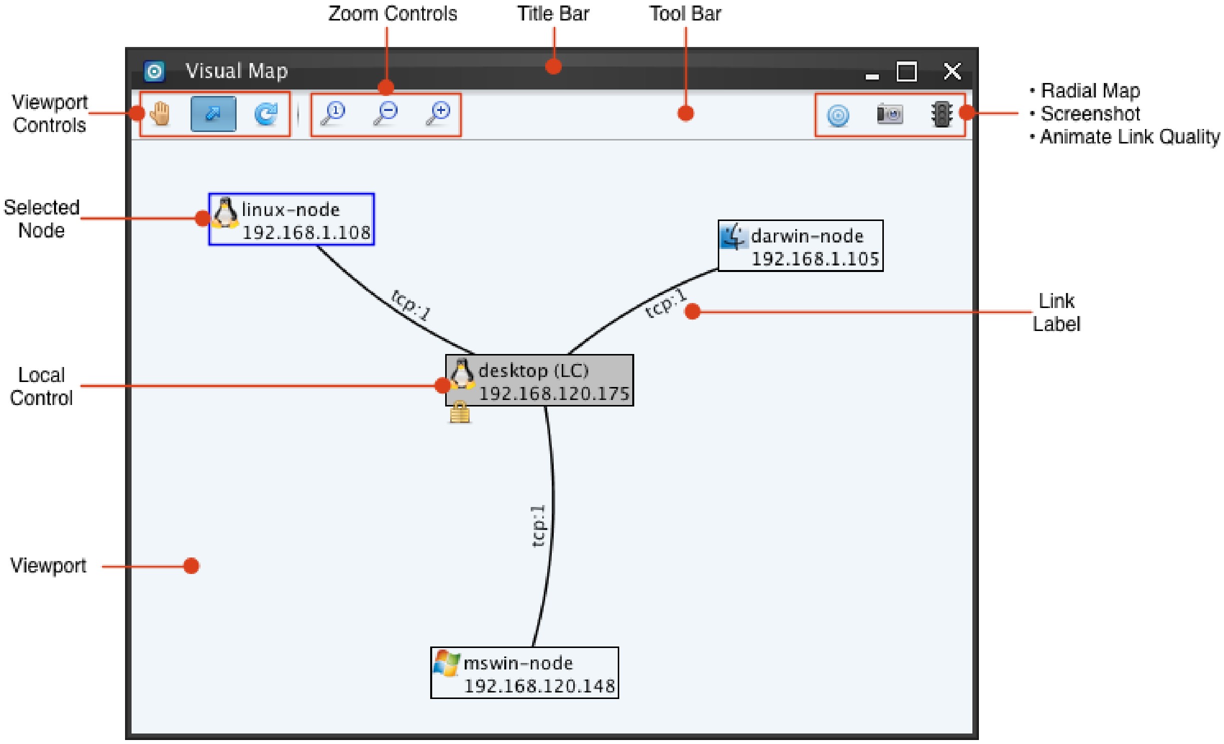


Fig. 5.35: **Visualization Map Window**

5.6.2 Navigating the Visual Map

Pan

The "Pan" button changes the mousing mode to panning. In panning mode the mouse can be dragged on the viewport to navigate around the visualization. This is useful if some of the devices are currently not visible in the viewport.

Pick

The "Pick" button changes the mousing mode to select and reposition devices. In pick mode the mouse can be pressed over a devices' icon and dragged to carry the device to a new location. Edges stretch as the device is dragged. Edges cannot be manually repositioned.

Refresh Layout

The "Refresh Layout" button attempts to do a smart layout of all the devices in the visualization. All devices with unlocked locations (*Locking and Unlocking a Device*) are distributed automatically to reduce cluttering and overlapping of devices and edges. The graph will perform a refresh automatically when devices are added or removed. It will attempt to maximize usage of available space in the window.

Zoom Actual

The "Zoom Actual" button resets the zoom level. The button can help locate the graph if one has zoomed too far in or out and lost their place in the visualization.

Zoom In

The "Zoom In" button zooms the viewport of the visualization in. The actual size of icons and text labels do not scale in size; zooming in allows more viewable space for the edges. Usually the "Pan" button is used after zooming in to move the viewport around.

Zoom Out

The "Zoom Out" button zooms the viewport of the visualization out. By default, the visualization is already zoomed to a one-to-one ratio with the size of the window when Horizon starts up.

Radial Map

The "Radial Map" button changes the layout of the map to a radial layout. The currently selected device will become the central device in the map's center. If no device is selected the Local Control becomes the central device. Radial Maps show the minimum distance between devices in number of link hops. Devices placed in the inner most ring are one hop distance, which means they have a direct link to the central device in the map. The next inner most ring displays devices that have a minimum path of two links to traverse to reach the central device and so on. To switch back to the original layout click the "Linear Map" button in the Tool Bar.

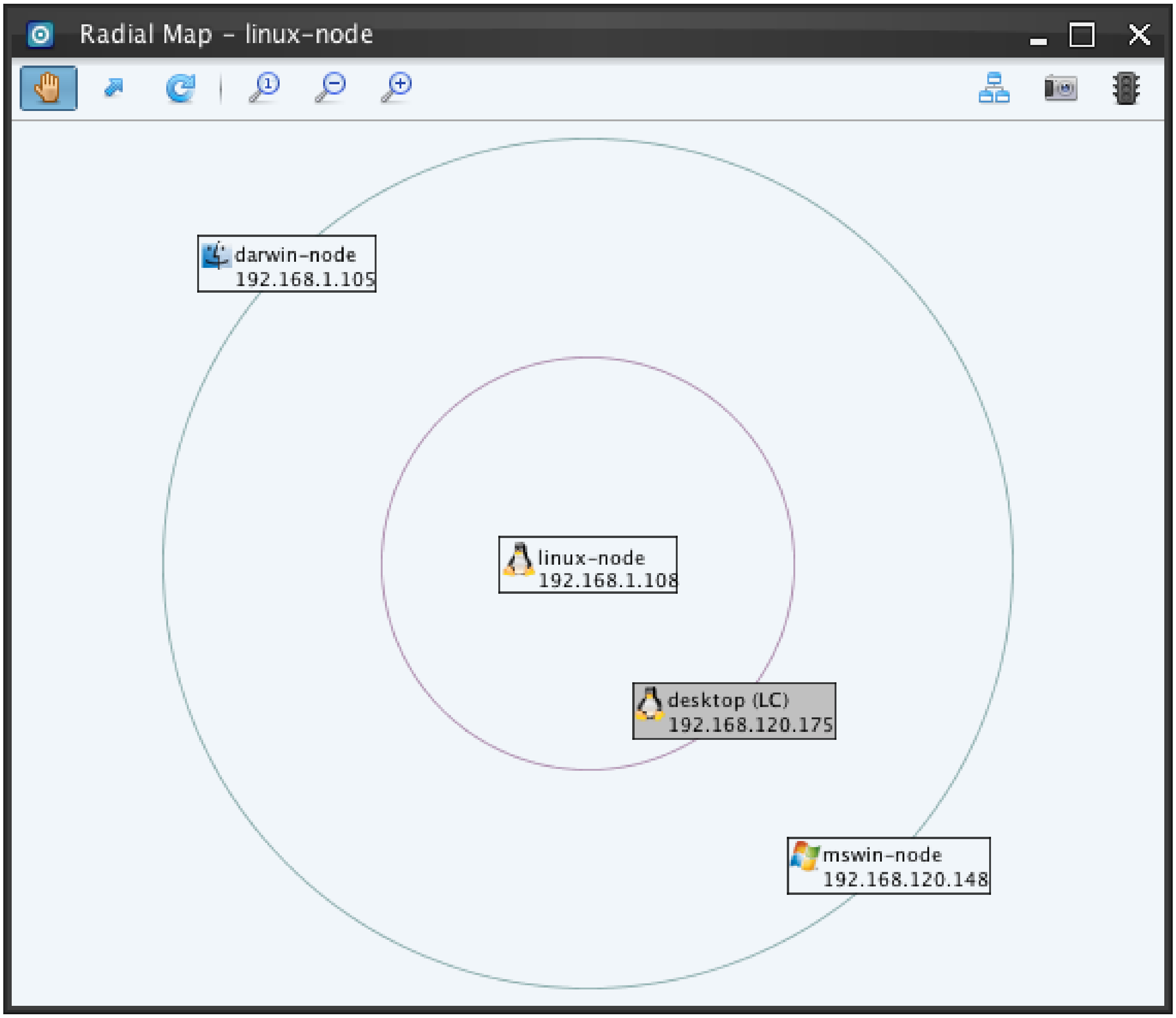


Fig. 5.36: **Radial Layout of Visual Map**

Screenshot

📷 The “Screenshot” button captures the entire Visual Map in either Linear or Radial Map mode and saves the image to a file. The image created will be large enough to show all devices in currently connected.

Animate Link Quality

🚦 Operators can toggle the “Animate Nodes” button to enable or disable a visual warning of devices with poor connectivity in the mesh. When the feature is enabled the button shows a lit traffic light 🚦. When a device’s packet loss or round trip time exceeds warning or critical thresholds the entry begins to pulse yellow or red respectively. The threshold for packet loss and round trip time can be set in the Device Preferences panel of the *Device Window*.

Resizing

The Visual Map Window can be resized. Resizing the window adjusts the overall size of the visualization along with the viewport. Clicking the “Refresh Layout” button 🔄 after resizing causes the layout to spread out or contract.

Network Menu

Operators can select the “Network” submenu to navigate the same right-click menu options from the *Network Window*‘s Network List.

Locking and Unlocking a Device

Operators can right click on a device and select the “Lock” menu item to pin the device to the location, prohibiting the “Refresh Layout” button from moving the device. Once locked, 🔒 appears at the lower left of the device’s label. The “Unlock” menu item undoes this operation.

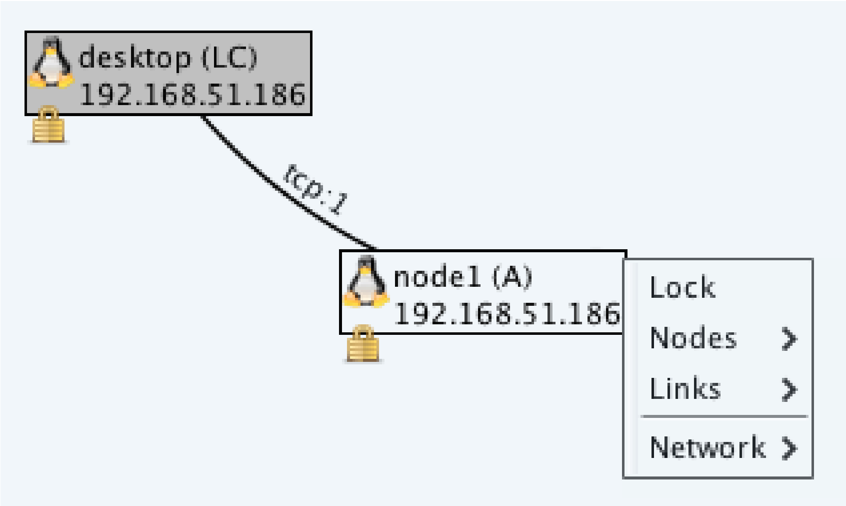


Fig. 5.37: **Nodes Locked on Visual Map**

Customizing Nodes

Operators can right click on a device and toggle menu items under the "Node" menu to customize the amount of information presented in the Visual Map.

- Abbreviated Name: Displays only first few characters from the Device Name followed by a horizontal ellipsis.
- Full Name: Displays the full Device Name. This items toggles with "Abbreviated Name".
- Alarm Status: Displays the Shutdown Alarm Status as described in *Shutdown Alarm State Table*.
- Control Status: Displays the Control Status as described in *Control Status Table*.
- IP Address: The host's IP address associated to one of the links of the device. This may or not be the external outward facing IP address when multiple links exist for the device.
- OS Icon: Displays an icon representing the operating system from the *Operating System Icons Table*.
- Packet Loss: Displays packet loss formatted as "PL:##".
- Round Trip Time: Displays the round trip time in milliseconds formatted as "RT:##".

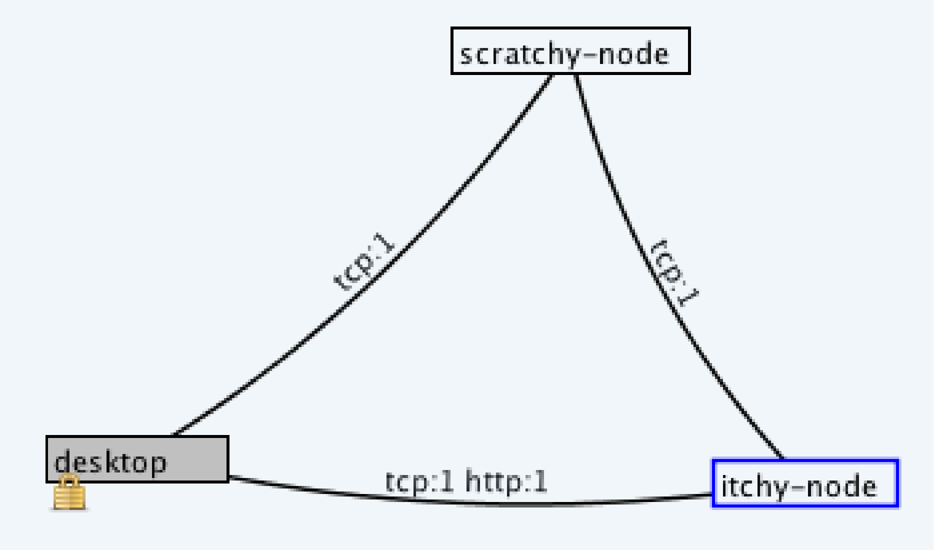


Fig. 5.38: **Minimal Device Labels on Visual Map**

Customizing Links

Operators can right click on a device and toggle menu items under the "Links" menu to customize the amount of information presented for links in the Visual Map.

- Show Link Labels: Enables and disables labels on links. When labels are disabled the link information can still be accessed by hovering the mouse over the link to produce the tool tip.
- Parallel Links: Expands and collapses edge lines to represent parallel links. When in parallel link mode, if devices have multiple links between them an edge is drawn on the graph to represent each individual link. When parallel link mode is disabled one edge represents all links between the nodes and the link label displays counts of all link types between the nodes.
- Link Type: Requires Parallel Links mode to be enabled. This menu item displays the link type on each link between the nodes.

- Ports: Requires Parallel Links mode to be enabled. This menu item displays the local port numbers for each side of the link in the format of “### : ###”

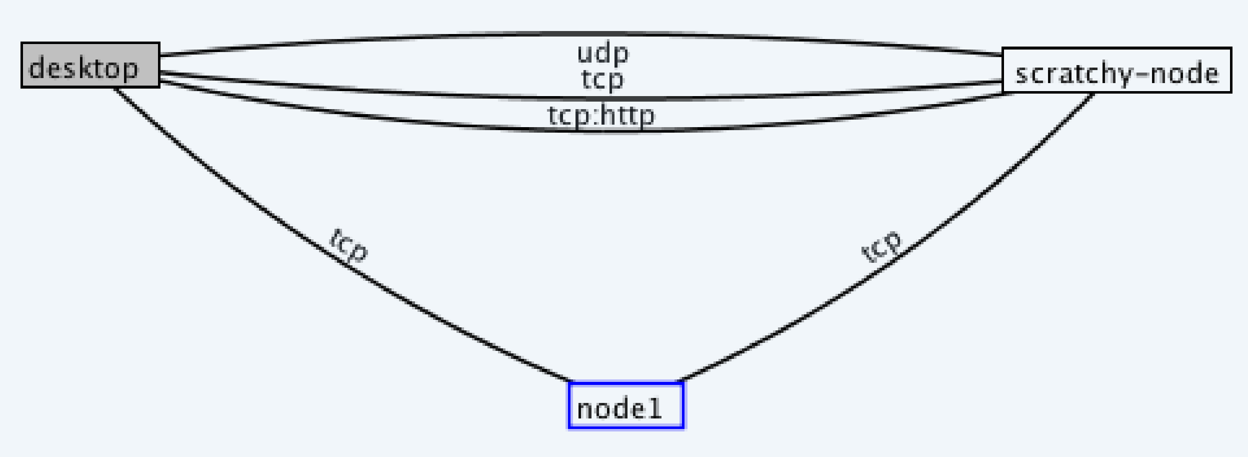


Fig. 5.39: **Parallel Links on Visual Map**

5.7 IP Version 6 Support

Release version 3.4 added support for IPv6 service ports and links. Service Port and Link Configurations have an IP Version combo box. For backward compatibility, both service and link default to IP version 4.

5.7.1 Address Notations

IPv6 uses colon as a delimiter instead of period. IPv4 addresses use colon as a separator from a remote port. IPv4 address appear in the notation: xxx.xxx.xxx.xxx:port. The Link Table surrounds IPv6 addresses with square brackets to make IPv6 addresses distinguishable from their port, [xxxx:xxxx::xxxx]:port

IPv6 allows for addresses to abbreviate one span of 0’s in the address with double colon, ::.

Links can show IPv4 addresses mapped to IPv6 mapped with a prefix of ::ffff:xxx.xxx.xxx.xxx. For example, 222.1.41.90 would would map to IPv6 ::ffff:222.1.41.90. Addresses may appear this way in the Link Table because of Network Address Translation or hostname resolution for IPv6 being given an IPv4 address.

In IPv4 0.0.0.0 is shown in the Service Table to indicate that the port is bound to all IP addresses for the local machine. In IPv6, the equivalent is double colon, :: Thus, an IPv6 or dual port in the Service Table is represented as ”:::port”.

Localhost IPv4 address is 127.0.0.1 and in IPv6 notation localhost is ::1.

The Link Panel supports IPv6 link local addresses as well. Link local addresses starts with the network prefix: **fe80**. To originate a link using a link local address a network interface must be specified. IPv6 supports this by using a special character % followed by the interface descriptor (scope id). For example, a valid link local address can be entered in the Link Panel’s Host field as fe80::6a5b:35ff:fe89:4931%eth1 to indicate the connection should connect using eth1 network adapter. The Link Panel and Network Window both hide the scope id by default, but it can be shown by toggling a checkbox in the Device Preferences of the *Device Window*.

5.7.2 Service Ports

The Service Panel’s configuration allows for IPv4 only (4), IPv6 only (6), or dual-stack (dual) under the IP Version combo box. When dual is used, incoming links can be accepted regardless of IP version.

Note: Some platforms do not support dual and some platforms will not support IPv6 only. Service port requests will fail on these platforms for the unsupported IP version and provide a message in Log Window. For example, Linux 32-bit does not support IPv6 only but does support IPv4 or dual.

5.7.3 Links

Link Panel's IP Version combo box, but is slightly different from service. Links allow the request to select IPv4 only (4), IPv6 only (6), or a hostname resolution preference for one over the other. Version preference requests are listed as "4,6" to prefer IPv4 hostname resolution over IPv6 hostname resolution. Likewise, "6,4" prefers IPv6 hostname resolution over IPv4 hostname resolution. When IPv4 only or IPv6 only is specified, the remote hostname will only be resolved using specified version. Remote hostname resolution always occurs on the machine the link is originating from.

Table 5.4: IP Version Service and Link Compatibility Table

Link Type	Remote Host Resolution	Link is Requested	Service 4	Service 6	Service dual
4	with hostname resolvable with IPv4	Y	Y	N	Y
4	with hostname not resolvable with IPv4	N	N	N	N
6	with hostname resolvable with IPv6	Y	N	Y	Y
6	with hostname not resolvable with IPv6	N	N	N	N
4,6	with hostname resolvable with IPv4	Y	Y	N	Y
4,6	with hostname resolvable with IPv6	Y	N	Y	Y
4,6	with hostname not resolvable with IPv4 or IPv6	N	N	N	N
6,4	with hostname resolvable with IPv4	Y	Y	N	Y
6,4	with hostname resolvable with IPv6	Y	N	Y	Y
6,4	with hostname not resolvable with IPv6 or IPv4	N	N	N	N

IPv6 Examples

Say "alpha.my-example.lan" is entered in Link Panel's Host field with IPv6 selected in the Version combo box. The node originating the link must be able to resolve the host locally or with its DNS to an IPv6 address for "alpha.my-example.lan".

For the below examples, assume is a **service port set to IPv6** only on the remote machine.

- Remote Host field is set to an IPv4 address and IP Version is set to 4. This fails because the link is attempting to connect to an IPv4 service port.

- Remote Host field is set to an IPv6 address and IP Version is set to 6. This succeeds because the link is attempting to connect to an IPv6 service port.
- Remote Host field is set to a hostname: "alpha.my-example.lan", the machine can resolve to an IPv4 address, and the IP Version is set to 4. This fails because the link is attempting to connect to an IPv6 service port.
- Remote Host field is set to a hostname: "alpha.my-example.lan", the machine can resolve to an IPv6 address, and the IP Version is set to 6. This link request succeeds.
- Remote Host field is set to a hostname: "alpha.my-example.lan" and IP Version is set to 4,6 (prefer IPv4). The link originating node first attempts to resolves the hostname to an IPv4 address. If it resolves to an IPv4 address, the link request is be for IPv4 and therefore the link request fails. However, if the originating host fails to resolve the hostname to an IPv4 address, the originating node tries to resolve the hostname to an IPv6 address **before** sending out a link request. If an IPv6 address resolves successfully, the node then issues an IPv6 link request and the link succeeds. The important concept to grasp is the preference **only** applies to hostname resolution, as only one link request is ever made. The link request is always made with the first resolving address.
- Remote Host field is set to a hostname: "alpha.my-example.lan" and IP Version set to 6,4 (prefer IPv6). This works the same as the above but with hostname resolution preference to IPv6. The link originating node tries to resolve the hostname to IPv6 address. If resolution succeeds, a link request is sent out and in this example the link request succeeds. If an IPv6 address is not resolved, the originating node tries to resolve to an IPv4 address. If an IPv4 address resolution also fails, no link request occurs. If it is successful, a link request is sent out. In this case, the link link request still fails because the remote service port is IPv6 only.

For the next example, assume there is a service port 5005 dual on the remote machine.

- Remote Host field is set to an IPv4 address 172.66.33.54 and IP Version is set to 4. The link succeeds. Also, most operating systems will choose to do IPv4 address mapping to IPv6 described above. Therefore, the Link Table on the remote node will display: [::ffff:172.66.33.54]:5005 by using square brackets around the IPv6 address to keep the port distinguishable.

5.8 HTTP(S) Server Response Options

Service ports for HTTP(S) include an HTTP protocol identifying header with some common attributes. The Server Header Index in the Service options dialogs for HTTP(S) service ports allows users to select from a few different options. The default is always index 0. There are four for Microsoft Windows nodes and twelve for Posix OS. If an out of bound index is selected, the default, zero index, is used. The table below lists the attributes and values for each index.

Note: "Content-Type" values will be text/html, image/png, or gzip depending upon the connection request type.

Table 5.5: Microsoft Windows HTTP Server Header Indices

Index #	HTTP Field	HTTP Values
0	Cache-Control	no-cache, private, no-store, must-revalidate
	Content-Length	varies
	Content-Type	text/html; charset=utf-8, image/png, or gzip
	Server	Microsoft-IIS/8.5
	Date	present date & time
1	Cache-Control	no-cache, private, no-store, must-revalidate
	Content-Length	varies
	Content-Type	text/html; charset=utf-8, image/png, or gzip
	Server	Microsoft-IIS/7.5
	X-Powered-By	ASP.NET
	Date	present date & time
2	Cache-Control	no-cache, private, no-store, must-revalidate
	Content-Type	text/html; charset=utf-8, image/png, or gzip
	Server	Microsoft-IIS/7.5
	X-Powered-By	ASP.NET
	Date	present date & time
	Connection	keep-alive
	Content-Length	varies
3	Cache-Control	no-cache, private, no-store, must-revalidate
	Content-Type	text/html; charset=utf-8, image/png, or gzip
	Server	Microsoft-IIS/7.5
	X-Powered-By	ASP.NET
	X-UA-Compatible	IE=edge
	Date	present date & time
	Content-Length	varies

Table 5.6: POSIX HTTP Server Header Indices

Index #	HTTP Field	HTTP Values
0	Server	nginx
	Date	present date & time
	Content-Type	text/html;charset=UTF-8, image/png, or gzip
	Content-Length	varies
	Connection	keep-alive
	Cache-Control	no-cache, private, no-store, must-revalidate
1	Content-Type	text/html, image/png, or gzip
	Accept-Ranges	bytes
	Content-Length	varies
	Connection	keep-alive
	Date	present date & time
	Server	lighttpd/1.4.32
	Cache-Control	no-cache, private, no-store, must-revalidate
2	Server	Resin/4.0.28
	Content-Type	text/html, image/png, or gzip
	Cache-Control	no-cache, private, no-store, must-revalidate

Continued on next page

Table 5.6 – continued from previous page

Index #	HTTP Field	HTTP Values
	Content-Length	varies
	Accept-Ranges	bytes
	Date	present date & time
	Connection	keep-alive
3	Server	nginx
	Content-Type	text/html; charset=utf-8, image/png, or gzip
	X-UA-Compatible	IE=Edge,chrome=1
	Content-Length	varies
	Cache-Control	no-cache, private, no-store, must-revalidate
	Date	present date & time
	Connection	keep-alive
4	Server	nginx
	Date	present date & time
	Content-Type	text/html; charset=utf-8, image/png, or gzip
	Content-Length	varies
	Connection	keep-alive
	Cache-Control	no-cache, private, no-store, must-revalidate
	X-Powered-By	PHP/5.4.4-14
5	Server	nginx/1.4.4
	Date	present date & time
	Content-Type	text/html; charset=UTF-8, image/png, or gzip
	Content-Length	varies
	Connection	keep-alive
	X-Powered-By	PHP/5.3.3-7
	Cache-Control	no-cache, private, no-store, must-revalidate
6	Server	Apache
	Content-Type	text/html; charset=UTF-8, image/png, or gzip
	Cache-Control	no-cache, private, no-store, must-revalidate
	Date	present date & time
	Connection	keep-alive
	Content-Length	varies
7	Date	present date & time
	Server	Apache/2.2.23 (Unix) mod_ssl/2.2.23 OpenSSL/0.9.8m
	X-Powered-By	Phusion Passenger (mod_rails/mod_rack) 3.0.15
	X-UA-Compatible	IE=Edge,chrome=1
	Cache-Control	no-cache, private, no-store, must-revalidate
	Content-Length	varies
	Content-Type	text/html; charset=utf-8, image/png, or gzip
8	Date	present date & time
	Server	Apache/2.2.22 (Ubuntu)
	Cache-Control	no-cache, private, no-store, must-revalidate
	Content-Type	text/html; charset=utf-8, image/png, or gzip
	Content-Length	varies
	Connection	keep-alive
9	Date	present date & time
	Server	Apache/2.2.3 (CentOS)
	Accept-Ranges	bytes
	Content-Length	varies
	Cache-Control	no-cache, private, no-store, must-revalidate

Continued on next page

Table 5.6 – continued from previous page

Index #	HTTP Field	HTTP Values
	Connection	keep-alive
	Content-Type	text/html; charset=utf-8, image/png, or gzip
10	Cache-Control	no-cache, private, no-store, must-revalidate
	Content-Type	text/html; charset=utf-8, image/png, or gzip
	Date	present date & time
	Pragma	no-cache
	Server	Apache
	Content-Length	varies
	Connection	keep-alive
11	Server	Apache
	Content-Type	text/html; charset=utf-8, image/png, or gzip
	Pragma	no-cache
	Date	present date & time
	Content-Length	varies
	Connection	keep-alive
13	Server	Apache/2.2.17
	Content-Type	text/html; charset=utf-8, image/png, or gzip
	Content-Language	en-GB
	Date	present date & time
	Content-Length	varies
	Connection	keep-alive
	Cache-Control	no-cache, private, no-store, must-revalidate

CHAPTER

SIX

DISK MODULE

The Disk Module provides a range of features that include but are not limited to browsing a file system, securely deleting files, and transferring files. The features are divided into three broad categories: Disk Window, Directory Listing, and Transfers.

6.1 Disk Window

The Disk Window is a global window accessible from the Dashboard's Module combo box when the Local Control has the Disk Module activated. This window provides an operator with the Disk Preferences, Transfer History, and Directory Walk panels. An operator can open this window by selecting the Disk option in the Module Combo Box:

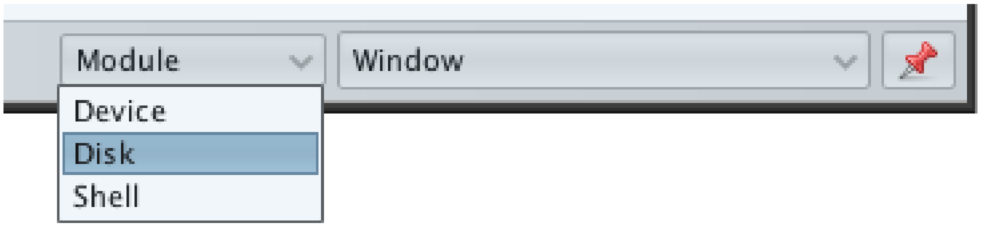


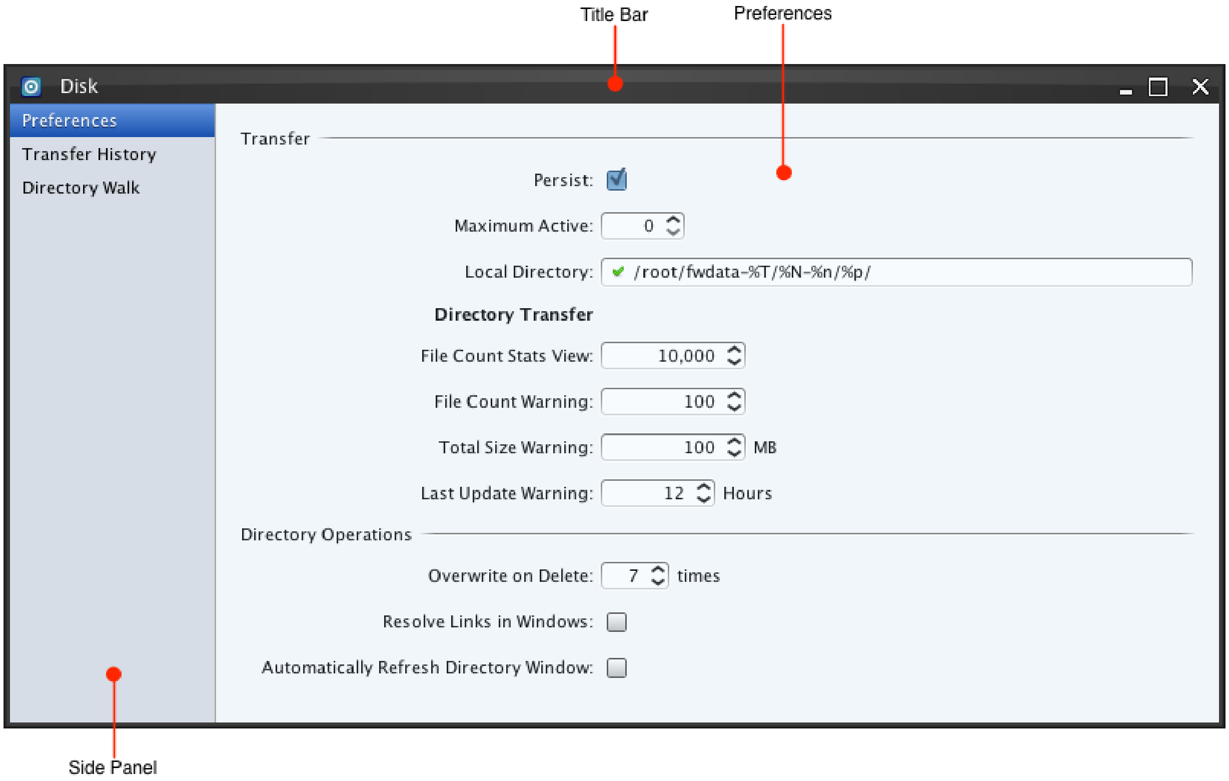
Fig. 6.1: **Dashboard Module Menu**

6.1.1 Preferences

Opening the Disk Window and selecting the Preferences option in the side panel displays the Disk Preferences panel.

The Disk Preferences panel consists of the following fields:

- Transfer
  - Persist: File transfers will persist in the event the originating Control is shutdown.
  - Maximum Active: Maximum number of active file transfers initiated from the Local Control at a time.
    - 0: distribute all file transfers to their respective Node/Control immediately.
    - >0: distribute a maximum of X file transfers at a time.
  - Local Directory: Directory on the Local Control where files will be stored when using the Download file transfer option. Path format options:
    - %n unique stack ID



Transfer

Persist: ☑

Maximum Active: 0

Local Directory: ✓ /root/fwdata-%T/%N-%n/%p/

Directory Transfer

File Count Stats View: 10,000

File Count Warning: 100

Total Size Warning: 100 MB

Last Update Warning: 12 Hours

Directory Operations

Overwrite on Delete: 7 times

Resolve Links in Windows: □

Automatically Refresh Directory Window: □

Fig. 6.2: **Preferences Panel**

        - %N operator defined Node name in Horizon
        - %t current time
        - %T local module load time
        - %p full path from upload Node
        - %P drive letter (windows only)
    - Directory Transfer
        - File Count Stats View: Only display file stats if total file count exceeds this value.
        - File Count Warning: Display warning if file count exceeds this value when a transfer is executed.
        - Total Size Warning: Display warning if the total amount of data exceeds this value (Mb) when a transfer is executed.
        - Last Update Warning: Display warning if 1 or more files last update time is older than value set (hours) when a transfer is executed.
- Directory Operations
    - Overwrite on Delete: Number of times a file will be overwritten before it is deleted.
    - Resolve Links in Windows: Attempt to resolve file links on Nodes executing on Microsoft Windows.
    - Automatically Refresh Directory Window: Refresh directory window when all file operations have completed if the file operations affect the current directory.

Note: A directory will not be refreshed automatically if the only file operations executed in the current directory are *Hash* operations. A hash file operation does not change a file's characteristics and would not affect the operator's view of the current directory.

6.1.2 Transfer History

Opening the Disk Window and selecting the "Transfer History" option in the Side Panel displays the Transfer History panel.

The Transfer History panel consists of a Tool Bar and the Transfer History table. The Tool Bar provides the ability to clear 🧹 inactive file transfers, cancel all ✖ file transfers, and a Filter text field where keywords can be entered, potentially limiting the number of results shown in the Transfer History Table. The search engine processes keywords in real-time. The second component of this panel, Transfer History Table, is responsible for displaying detailed transfer history information about queued, active, paused, cancelled, failed, and completed file transfers. The Transfer History table is organized into eleven columns (nine visible by default) that describe the following:

    - • (Status): File transfer states through the following icons.

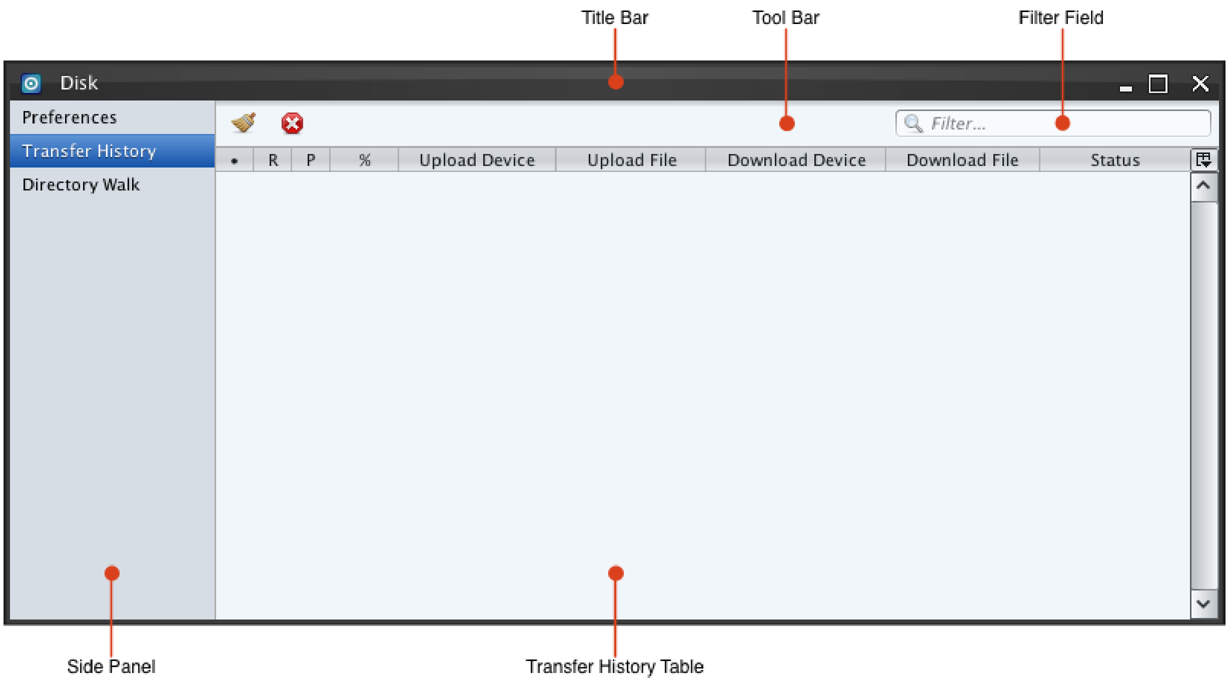


Fig. 6.3: **Transfer History Panel**

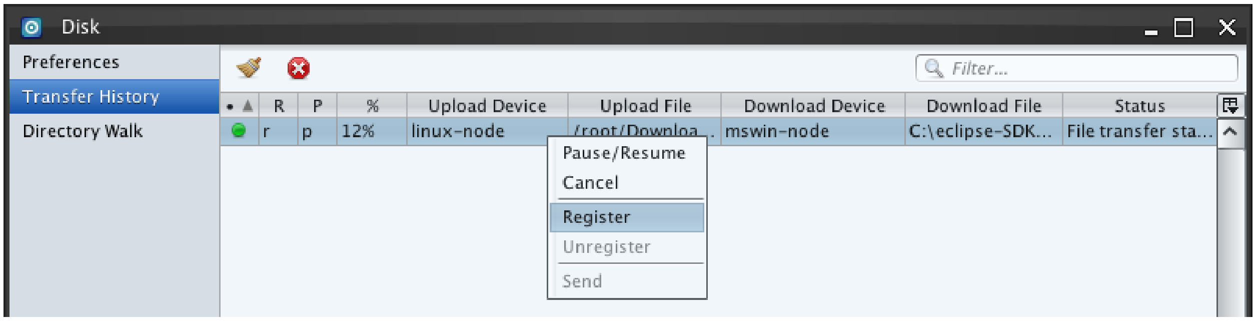
Symbol	Meaning
⌛	Local Control Queue
🟡	Queue
🟢	Active
⏸	Pause
⚪	Complete
🔴	Fail
✖	Cancel

Note: The maximum number of active file transfers allowed per Node is 5. As new file transfer requests arrive they are queued on the Node and processed on a first come first serve basis. Each Node/Control contains a single file transfer queue that is used by all other Nodes/Controls in the mesh network. This means that file transfer requests are processed in the order they are received. In order to control how many file transfers are sent to each Node/Control, a Local Control Queue has been implemented. This helps control the number of active file transfers on the mesh network originating from that Local Control.

- R: Registry option indicates the type of status updates the file transfer will receive.

Symbol	Meaning
-	Local Control started the file transfer and will receive all updates including start/stop/incremental/cancel/fail.
r	Local Control either did not start the file transfer or was restarted and received this file transfer during a sync. Only start/stop/cancel/fail updates will be received.
R	Local Control either did not start the file transfer or was restarted and received this file transfer during a sync. All updates will be received including start/stop/incremental/cancel/fail.

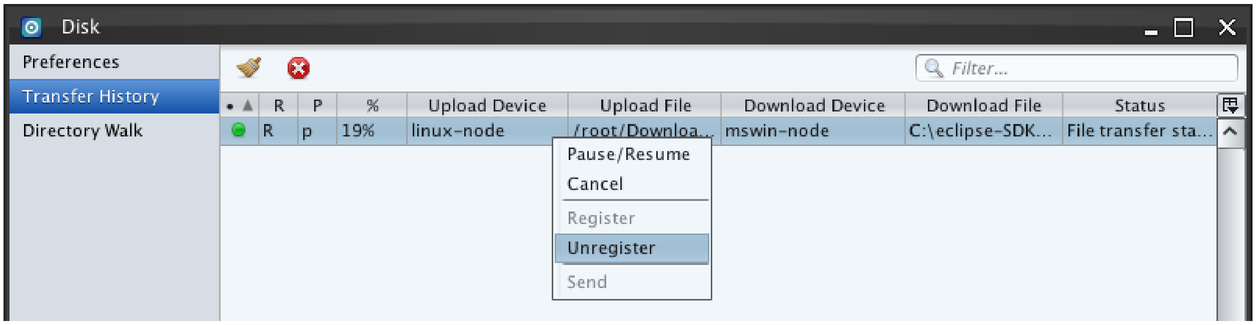
– File transfers that contain the registry option ‘r’ can be upgraded to receive all updates including start/stop/incremental/cancel/fail. This is done by right clicking on the file transfer and selecting the Register option.



• ▲	R	P	%	Upload Device	Upload File	Download Device	Download File	Status
●	r	p	12%	linux-node	/root/Downloa...	mswin-node	C:\eclipse-SDK...	File transfer sta...

Fig. 6.4: **Transfer History Menu - Register Option**

– File transfers that contain the registry option ‘R’ can be downgraded to receive only some updates including start/stop/cancel/fail. This is done by right clicking on the file transfer and selecting the Unregister option.



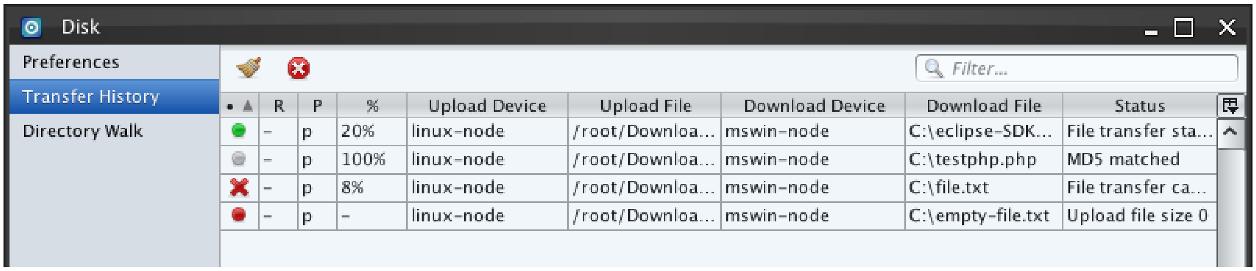
• ▲	R	P	%	Upload Device	Upload File	Download Device	Download File	Status
●	R	p	19%	linux-node	/root/Downloa...	mswin-node	C:\eclipse-SDK...	File transfer sta...

Fig. 6.5: **Transfer History Menu - Unregister Option**

- P: Persist option indicates whether a file transfer will continue if the originating Control is shutdown. This option is read from the Preferences panel when a new file transfer is started and cannot be changed.

Symbol	Meaning
-	Terminate the file transfer if the Control that started it is shutdown.
p	Continue the file transfer if the Control that started it is shutdown.

- Percent: Completed percent for the file transfer.
- Upload Device: Node or Control name where the file transfer is coming from.
- Upload File: Full path, including file name, where the file is being read.
- Download Device: Node or Control name where the file transfer is going.
- Download File: Full path, including file name, where the file is being written.
- Size (bytes): File size in bytes.
- Bytes (read/written): Real-time value of the number of bytes read/written.
- Status: String displaying the file transfer's current status.



• ▲	R	P	%	Upload Device	Upload File	Download Device	Download File	Status
🟢	–	p	20%	linux-node	/root/Downloa...	mswin-node	C:\eclipse-SDK...	File transfer sta...
⚪	–	p	100%	linux-node	/root/Downloa...	mswin-node	C:\testphp.php	MD5 matched
✖	–	p	8%	linux-node	/root/Downloa...	mswin-node	C:\file.txt	File transfer ca...
🔴	–	p	–	linux-node	/root/Downloa...	mswin-node	C:\empty-file.txt	Upload file size 0

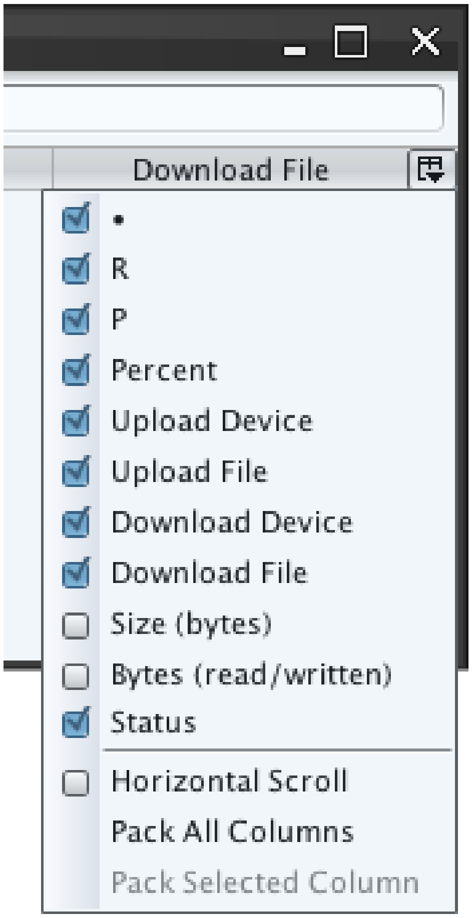
Fig. 6.6: **Transfer History Panel with File Transfers**

The Transfer History table can be tailored to display more information normally hidden from the operator. Clicking the icon in the top right of the table will bring up a menu that allows for columns to be hidden or unhidden along with other features. By default the Size (bytes) and Bytes (read/written) columns are hidden. These columns can be enabled by clicking on the appropriate row in the menu, indicated by a check mark next to the text. To hide a column, select the column name in the menu and the column will be hidden. Other options that can be enabled or disabled include Horizontal Scroll, Pack All Columns, and Pack Selected Column. The horizontal scroll bar provides horizontal viewing capabilities for columns out of view. The Pack All Columns and Pack Selected Column turn on word wrapping which automatically wraps any strings that exceed the allotted column width.

6.1.3 Directory Walk

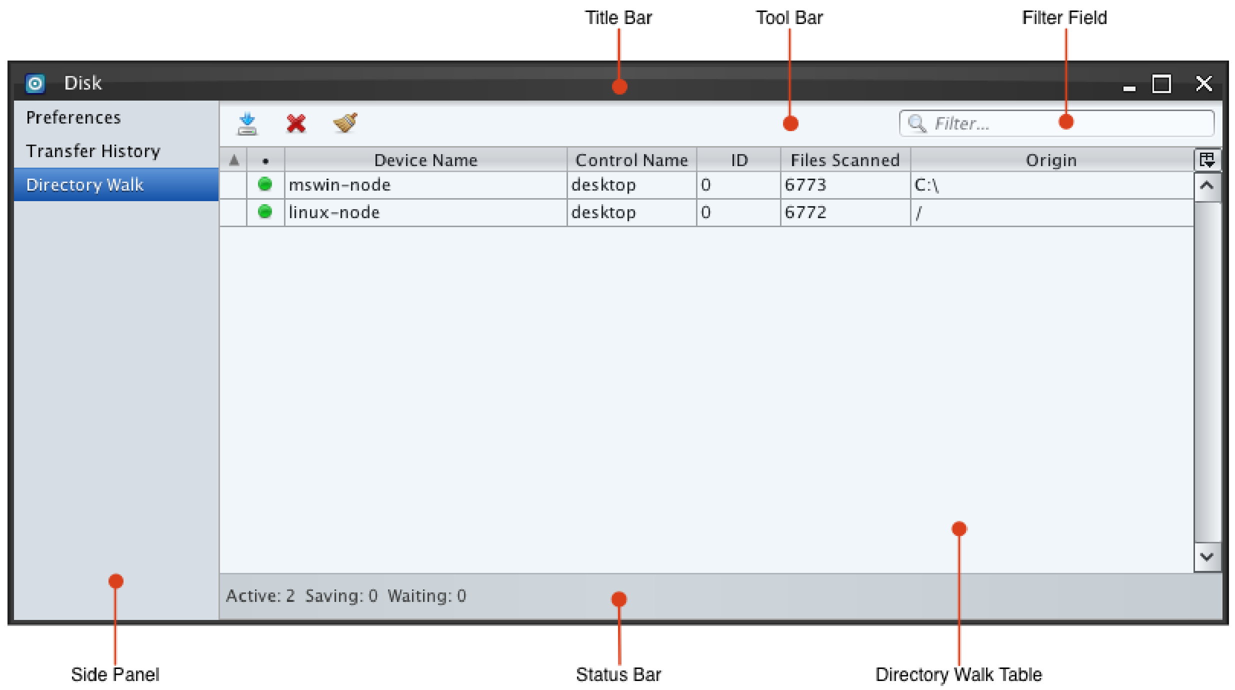
Opening the Disk Window and selecting the "Directory Walk" option in the Side Panel will display the Directory Walk panel:

The Directory Walk panel consists of a Tool Bar, Status Bar, and Directory Walk table. The Tool Bar provides a Search text field where keywords can be entered, restricting the results shown in the Directory Walk table. The Status Bar displays totals of the number of active directory walks, directory walks being saved, and directory walk requests waiting to begin. The Directory Walk table, is responsible for displaying status and statistic updates for directory



☑ •
☑ R
☑ P
☑ Percent
☑ Upload Device
☑ Upload File
☑ Download Device
☑ Download File
□ Size (bytes)
□ Bytes (read/written)
☑ Status
□ Horizontal Scroll

Fig. 6.7: **Transfer History Table Options**



Preferences
Transfer History
Directory Walk

▲	•	Device Name	Control Name	ID	Files Scanned	Origin
	🟢	mswin-node	desktop	0	6773	C:\
	🟢	linux-node	desktop	0	6772	/

Active: 2 Saving: 0 Waiting: 0

walks. Column headers can be clicked to customize live sorting based upon the column's content. A sort arrow indicates an ascending or descending sort. The sort arrow toggles when the column header is repeatedly clicked. The Directory Walk table is organized into seven columns that describe the following:

- Left Column: No Icon. This column allows sorting by the natural chronological ordering of the directory walk requests.
- • (Status): Directory walk states as follows:

Symbol	Meaning
🟡	Waiting. Request has been received by the Node but not started.
🟢	Active. Node is currently executing the walk request.
▤	Saving Results. Results of the walk are being saved to the Local Control's database.
⚪	Completed. Node has finished the request and the Local Control has saved the results.
🔴	Failed. Failures can include a Node being shutdown or the originating walk path being invalid.
✖	Canceled. Requests canceled by an operator.

- Device Name: The Node's name being walked.
- Control Name: The Control's name that issued the directory walk.
- ID: A unique directory walk identity number assigned by the Node. This number is only unique between directory walks conducted on the same Node.
- Files Scanned: The total number of files scanned by the walk.

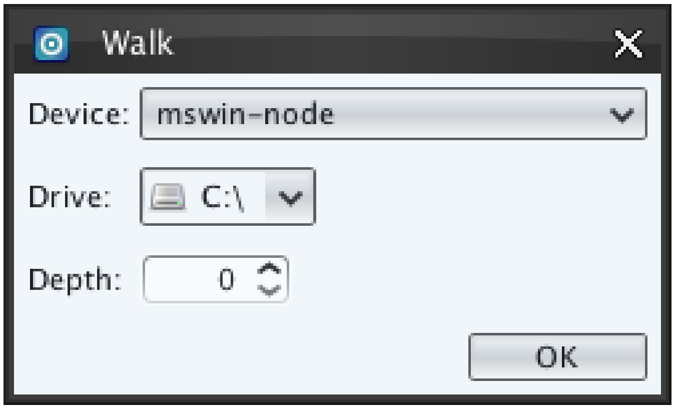
- Origin: The full directory path where the walk originates.

Note: Periodic updates of total files scanned and the “Saving Results” status are only sent to Horizon clients of the Control that issued the walk. Horizon clients of other Controls only receive major status changes: Waiting, Active, and finally Completed / Failed / Canceled.

Start Directory Walk

Press the “Start Directory Walk” button to begin a new walk. This brings up the Directory Walk dialog. The Walk Dialog allows operators to select how deep to recursively walk directories by adjusting the “Depth” spin box. By default, a depth of 0 performs a full walk of the file system. When a Microsoft Windows Node is specified in the Device combo box the Node scans for drives. The Walk Dialog displays the resulting Microsoft Windows drive letters and icons (indicate drive type).

Symbol	Meaning
▮	Removable disk such as a USB stick or floppy drive.
🖴	Hard disk drive.
💿	Optical disc drives.
▮	Network mounted disk/share.



Device: mswin-node

Drive: C:\

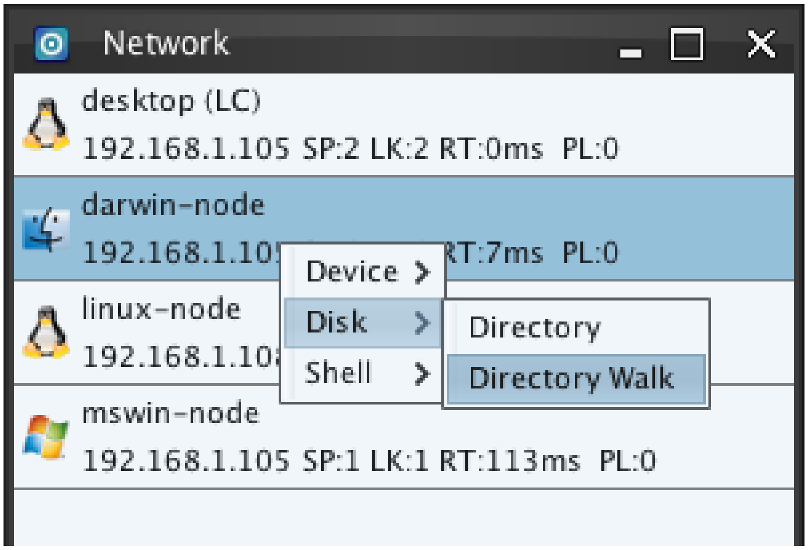
Depth: 0

Fig. 6.8: **Directory Walk Dialog**

Note: Microsoft Windows reports floppy drives and optical drives regardless of media presence. These drives report no contents when walked or read when empty.

Alternatively, the Directory Walk dialog appears by right clicking a Node in the Network Window and selecting “Disk” and “Directory Walk.”

In the Directory Walk dialog select the Node to walk from the combo box. The lower combo box will enable, allowing operators to select the drive or mount point to walk on the Node. Press the “Submit” button to issue the walk request. Once the Node receives the walk request it will issue a status update, which populates the Directory Walk table. The “Origin” column will display the drive or mount point selected.



desktop (LC)
192.168.1.105 SP:2 LK:2 RT:0ms PL:0

darwin-node
192.168.1.10 RT:7ms PL:0

linux-node
192.168.1.10

mswin-node
192.168.1.105 SP:1 LK:1 RT:113ms PL:0

Fig. 6.9: **Disk Menu - Directory Walk Option**

Note: Nodes only attempt to process a small number of walks at a time. Additional walk requests are queued, and the walk status indicates a "Waiting" ● status in the Directory Walk table. Likewise the number of walks a Horizon client can have outstanding is restricted to help prevent overloading the Local Control's database with an overwhelming amount of disk I/O activity.

Cancel Directory Walk

✖ A directory walk can be canceled by selecting its entry in the Directory Walk table while it displays "Waiting" ● or "Active" ● status indication and pressing the "Cancel Walk" button. Once the entry enters the "Saving Results" status the walk cannot be canceled. When active directory walks are canceled the results collected prior to cancelation are still saved to the database.

Clear Inactive Directory Walks

The "Clear Inactive Directory Walks" button removes all directory walk entries from the Directory Walk table that are finished. Entries marked as Canceled, Failed, or Completed are removed.

6.2 Directory Listing

Directory Listing provides a visual interface for a set of file system operations including browse, delete, get information, rename, make new directory, transfer, and find files and directories on both Nodes and Controls.

6.2.1 Target Node or Control

Browsing the file system of a Node or Control requires that a Node or Control be selected in the Network Window and the Directory option be chosen in the Disk menu.

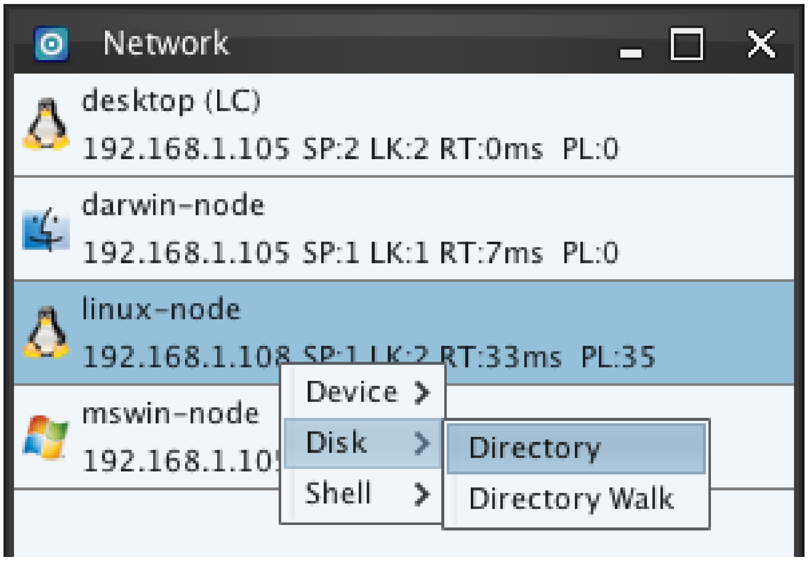

Network
desktop (LC)
192.168.1.105 SP:2 LK:2 RT:0ms PL:0
darwin-node
192.168.1.105 SP:1 LK:1 RT:7ms PL:0
linux-node
192.168.1.108 SP:1 LK:2 RT:33ms PL:35
mswin-node
192.168.1.10
Device
Disk
Shell
Directory
Directory Walk

Fig. 6.10: **Disk Directory Menu**

This will launch a Disk Directory window where the Node or Control's file system can be traversed once it is connected. The new Disk Directory window is also registered in the Windows combo box in the Dashboard. This provides an operator with a way to select a window and bring it to the front without having to search through all open windows. When a window is closed it is unregistered from the Windows combo box.

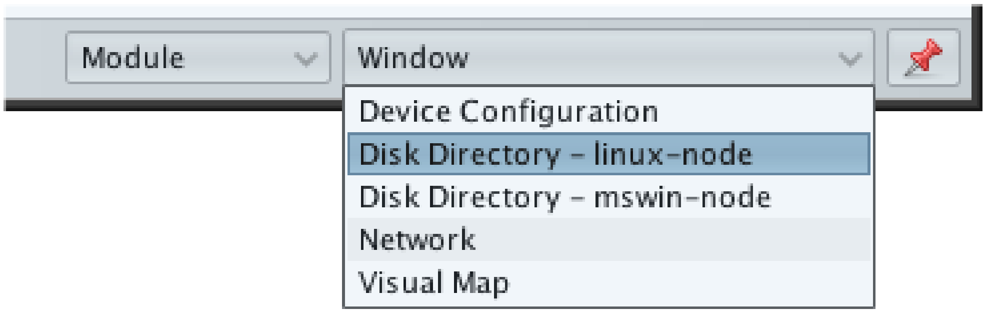


Fig. 6.11: **Dashboard Window Menu**

Each Disk Directory window consists of four components: Title Bar, Tool Bar, Directory Table, and Status Bar.

- Title Bar: Consists of the type of window (Disk Directory) along with the name of the Node or Control it is associated with. On the far right are a set of icons that control window specific operations, minimize, maximize, and close.
- Tool Bar: Provides an operator with the abilities to to connect, traverse, change file permissions, and search the current directory listing.
- Directory Table: Displays the current directory listing and is broken up into five columns:

  - • (File Type): Indicate the file type through one of two icons, file or directory. File icons are further broken into more specific icons depending on the file type (.doc, .png, .jpg, .pdf, etc.). A special status ⛔ displays in offline browsing mode to indicate that the directory entry has been removed from the current directory's path. This means it could have been deleted, renamed, moved, etc.
  - Name: File name sorted in alphabetical order.
  - Size: File size in bytes.
  - Mode: File type along with permissions. Unix file systems display the mode in the following format <file type><users><group><others> (i.e., drwxr–r–) while Windows uses the following format <file type><user> (i.e., -r-x).
  - Data Modified: Last time the file was modified.
- Status Bar: Provides information about the current directory (number of items and current path) along with status updates.

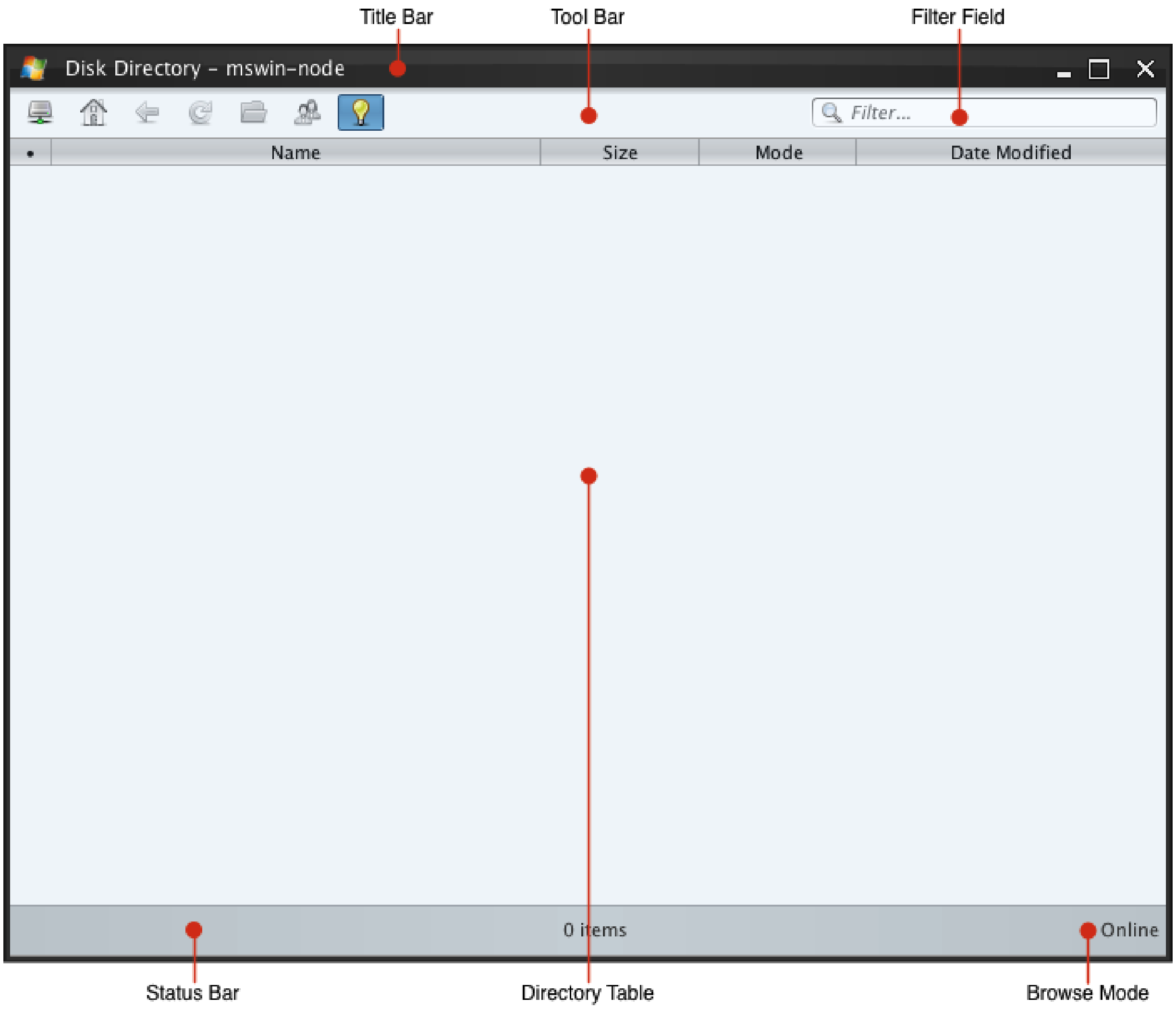
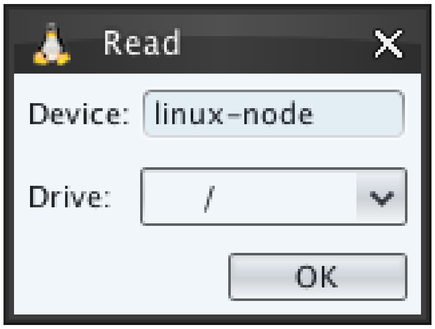


Fig. 6.12: **Disk Directory Window Overview**

Note: Disk Directory windows that are closed retain their information about the last directory traversed. This allows an operator to pick up where they left off in the event they reopen the Disk Directory window. If the Local Control is disconnected or a remote Node or Control that contained an active directory listing is shutdown the Disk Directory window will no longer be useable.

Connect

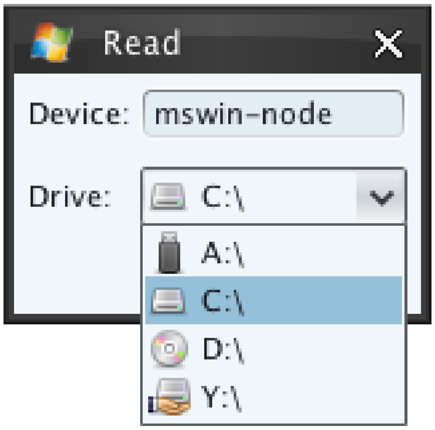
Browsing the file system of a Node or Control requires a drive connect action. The “Drive Connect” button will launch a Read dialog where the drive or mount point for the Node or Control can be selected. Unix devices make available a single mount point ‘/’ while Windows devices display drive letters and icons (indicate drive type).



Device: linux-node

Drive: /

Fig. 6.13: **Posix Drive Connect Dialog**



Device: mswin-node

Drive: C:\

A:\
C:\
D:\
Y:\

Fig. 6.14: **Windows Drive Connect Dialog**

Forward

Forward directory traversal is accomplished by double clicking on a directory.

Back

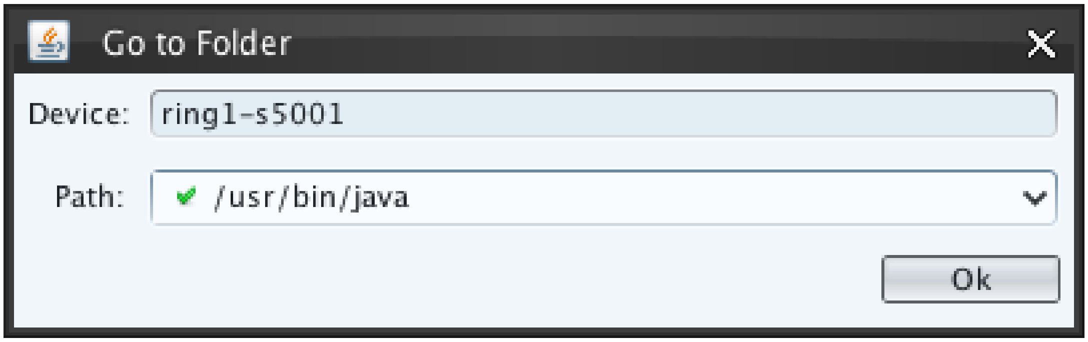
The “Back” button navigates up one directory to the directory from the current path.

Refresh

The “Refresh” button updates the current directory by refetching the contents. Directory contents are retrieved from the Node when in 💡 online mode and the Local Control’s database of when in 💡 offline mode. See *Browsing Switch* for more information on offline browsing.

Go To

The “Go To” button fetches the directory contents given a user supplied path. It allows intermediate directory navigation to be skipped. “No such file or directory” and “Unavailable” are displayed in online and offline modes respectively when an invalid path is entered.



Device: ring1-s5001

Path: ✓ /usr/bin/java

Fig. 6.15: **Go To Dialog**

Permissions

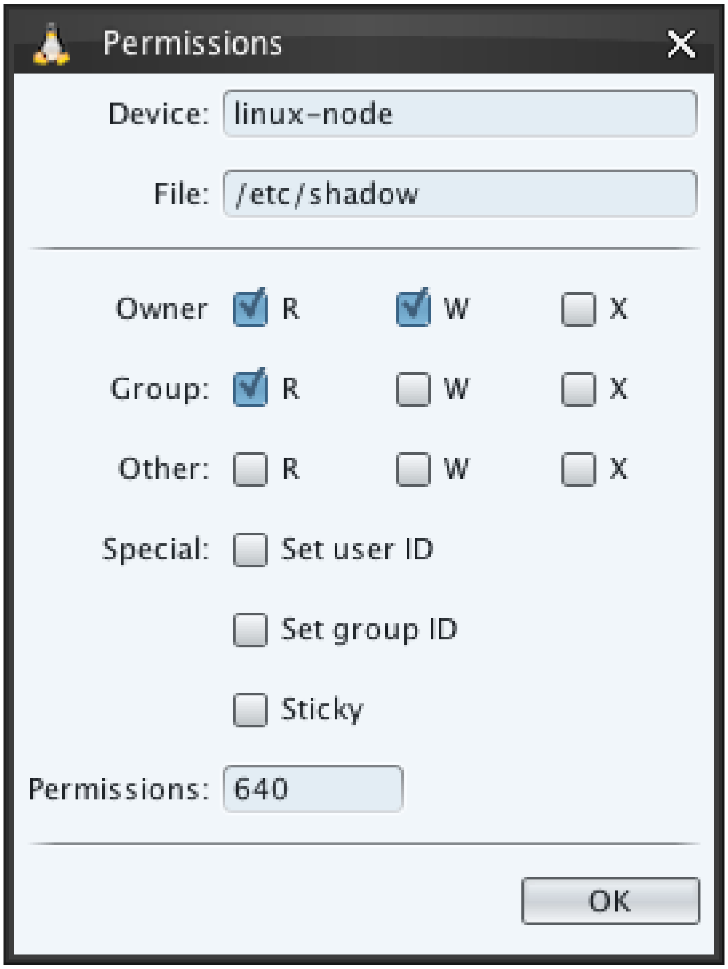
The “Permissions” button opens a dialog displaying the selected file or directory’s current permissions along with other available file permission options. These options include setting permissions for the Owner, Group, Other, and several other modes. All of these options apply to files residing in a Unix environment, while only the Owner permissions can be set on Windows files. If the selected file is a symbolic link on a Unix device, the specified permissions will apply to the link’s target file or directory.

The Permission dialog can also be launched by selecting a file or directory in the Directory Listing table and choosing the Get Info option in the right click menu.

Browsing Switch

💡 The Browsing Switch in the Tool Bar toggles the directory listing between online 💡 and offline 💡 browsing. Online browsing always requests directory listings from the Node. Offline browsing retrieves directory listing information from the Local Control’s database. Offline browsing utilizes results from completed directory walks and previous online browsing. Offline browsing mode is offered for Offline devices shown in the *Network Window*. The Browsing Switch is forced into 💡 for offline devices.

⛔ is displayed in the File Type column when an entry previously present in the current directory listing no longer does. This can be the result of the file being deleted, renamed, or moved. This icon is only displayed during offline browsing.



Permissions

Device: linux-node

File: /etc/shadow

Owner ☑ R ☑ W □ X

Group: ☑ R □ W □ X

Other: □ R □ W □ X

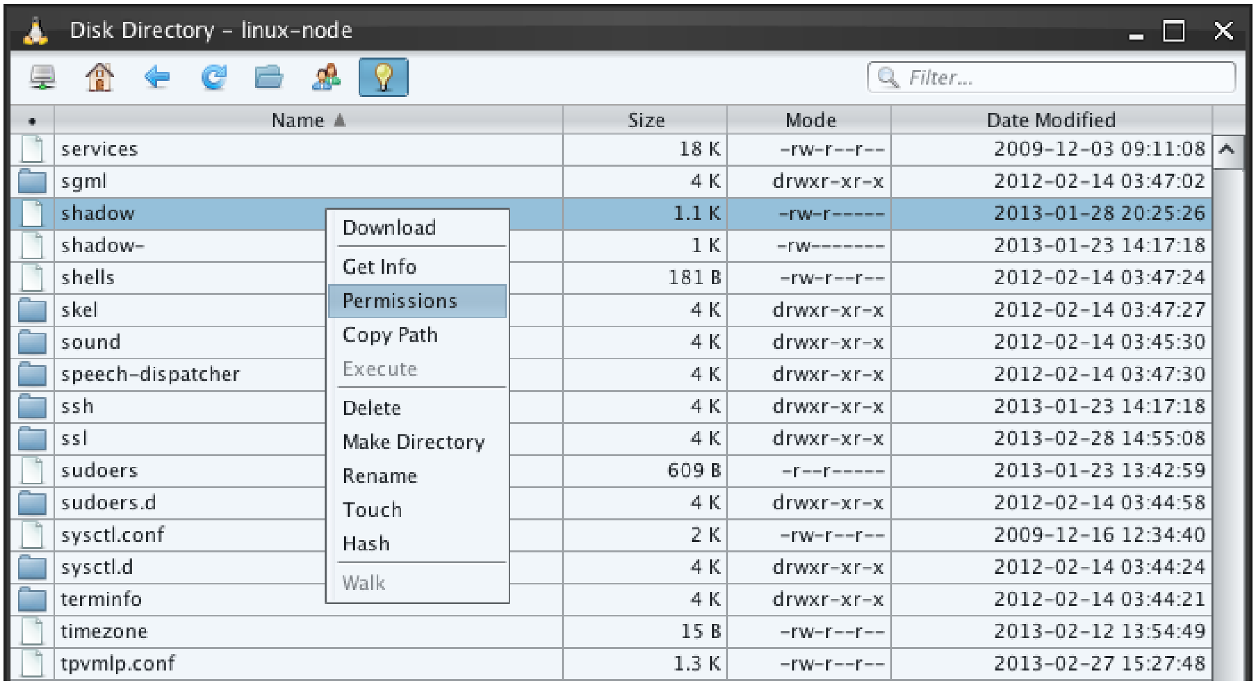
Special: □ Set user ID

□ Set group ID

□ Sticky

Permissions: 640

Fig. 6.16: **Permissions Dialog**



•	Name ▲	Size	Mode	Date Modified
📄	services	18 K	-rw-r--r--	2009-12-03 09:11:08
📁	sgml	4 K	drwxr-xr-x	2012-02-14 03:47:02
📄	shadow	1.1 K	-rw-r-----	2013-01-28 20:25:26
📄	shadow-	1 K	-rw-------	2013-01-23 14:17:18
📄	shells	181 B	-rw-r--r--	2012-02-14 03:47:24
📁	skel	4 K	drwxr-xr-x	2012-02-14 03:47:27
📁	sound	4 K	drwxr-xr-x	2012-02-14 03:45:30
📁	speech-dispatcher	4 K	drwxr-xr-x	2012-02-14 03:47:30
📁	ssh	4 K	drwxr-xr-x	2013-01-23 14:17:18
📁	ssl	4 K	drwxr-xr-x	2013-02-28 14:55:08
📄	sudoers	609 B	-r--r-----	2013-01-23 13:42:59
📁	sudoers.d	4 K	drwxr-xr-x	2012-02-14 03:44:58
📄	sysctl.conf	2 K	-rw-r--r--	2009-12-16 12:34:40
📁	sysctl.d	4 K	drwxr-xr-x	2012-02-14 03:44:24
📁	terminfo	4 K	drwxr-xr-x	2012-02-14 03:44:21
📄	timezone	15 B	-rw-r--r--	2013-02-12 13:54:49
📄	tpvmlp.conf	1.3 K	-rw-r--r--	2013-02-27 15:27:48

Fig. 6.17: **Directory Menu - Permissions Option**

Note: The only option available in the right click menu when using offline browsing is *File Information*. A restricted amount of information is captured during a Directory Walk, so less information may populate the Directory Listing during offline browsing. To retrieve the full properties, toggle the current directory online 💡, then press the 🔄 "Refresh" button.

While navigating in offline browsing mode, directories may display an "Unavailable" message. This message occurs in two circumstances: 1) the requested directory listing has not been walked or read through online browsing or 2) the requested directory listing is empty. To confirm an empty directory toggle back to online browsing 💡 mode.

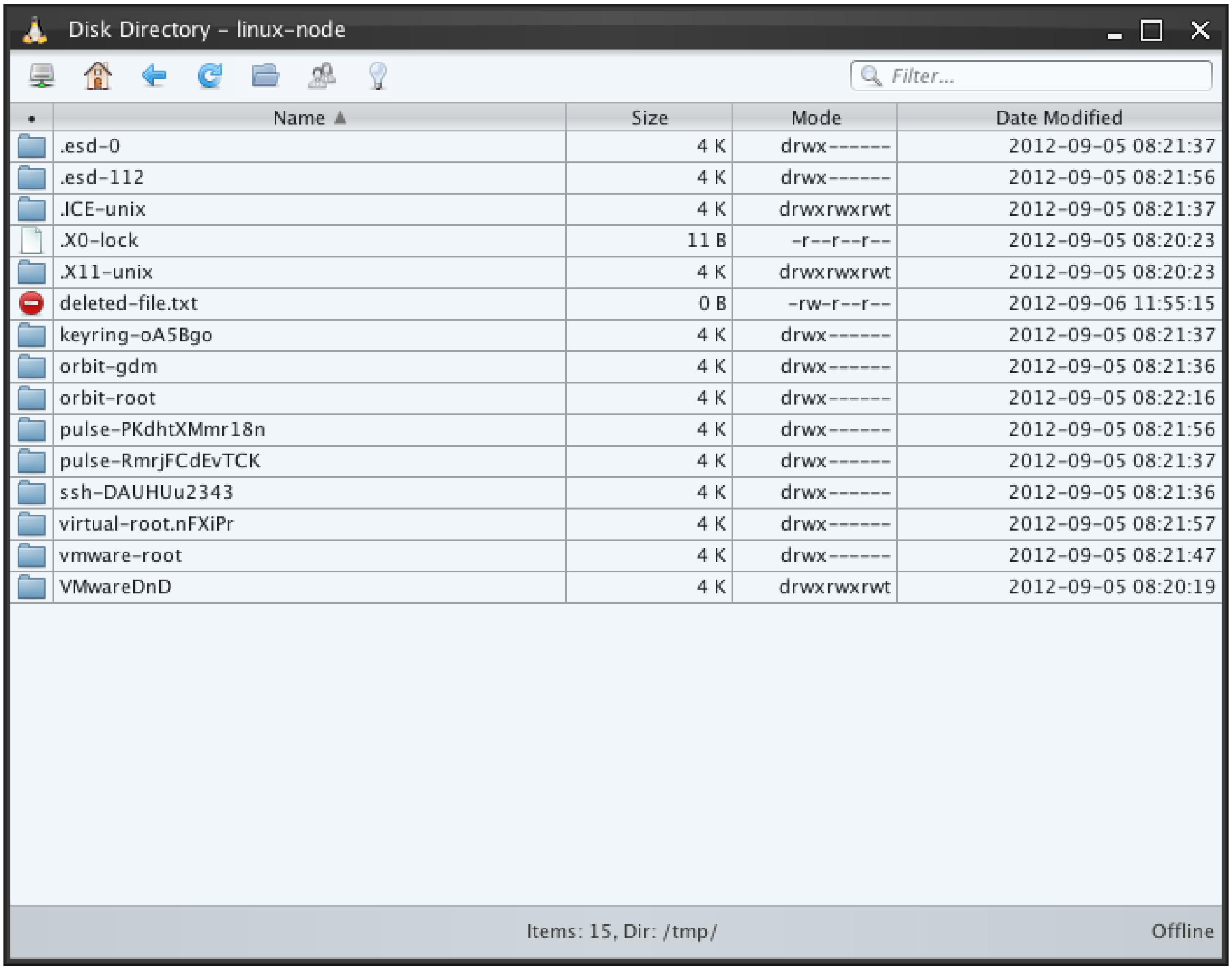
Search

The current directory listing can be filtered, effectively limiting the viewable results. Using the search text field in the Tool Bar, an operator can populate the text field with keywords. The Filter text field processes keywords in real-time by displaying any matching files, modes and dates. The keywords in the Search text field are removed when a connect, home, back, refresh, or a directory change occurs.

Links

Links are special files or directories created by the operating system that point to other files or directories. Links are displayed in the Disk Directory window with a unique icon. The following types of Links are supported:

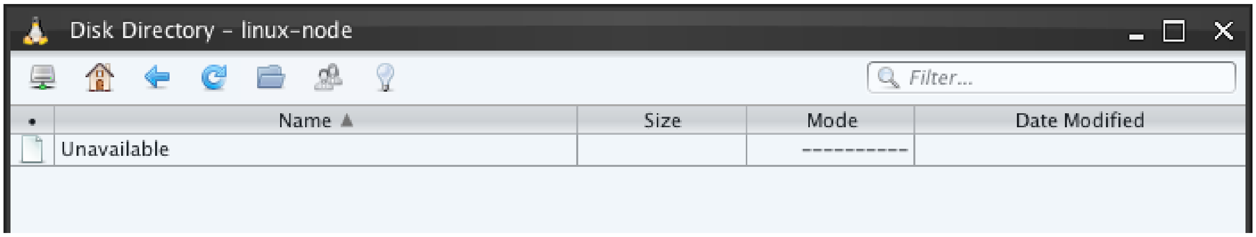
Type	Link is File or Directory	Operating System
Symbolic Link	File	Unix and Windows
Symbolic Link	Directory	Windows
Junction	Directory	Windows
.lnk Shortcut File	File	Windows



•	Name ▲	Size	Mode	Date Modified
	.esd-0	4 K	drwx------	2012-09-05 08:21:37
	.esd-112	4 K	drwx------	2012-09-05 08:21:56
	.ICE-unix	4 K	drwxrwxrwt	2012-09-05 08:21:37
	.X0-lock	11 B	-r--r--r--	2012-09-05 08:20:23
	.X11-unix	4 K	drwxrwxrwt	2012-09-05 08:20:23
	deleted-file.txt	0 B	-rw-r--r--	2012-09-06 11:55:15
	keyring-oA5Bgo	4 K	drwx------	2012-09-05 08:21:37
	orbit-gdm	4 K	drwx------	2012-09-05 08:21:36
	orbit-root	4 K	drwx------	2012-09-05 08:22:16
	pulse-PKdhtXMmr18n	4 K	drwx------	2012-09-05 08:21:56
	pulse-RmrjFCdEvTCK	4 K	drwx------	2012-09-05 08:21:37
	ssh-DAUHUu2343	4 K	drwx------	2012-09-05 08:21:36
	virtual-root.nFXiPr	4 K	drwx------	2012-09-05 08:21:57
	vmware-root	4 K	drwx------	2012-09-05 08:21:47
	VMwareDnD	4 K	drwxrwxrwt	2012-09-05 08:20:19

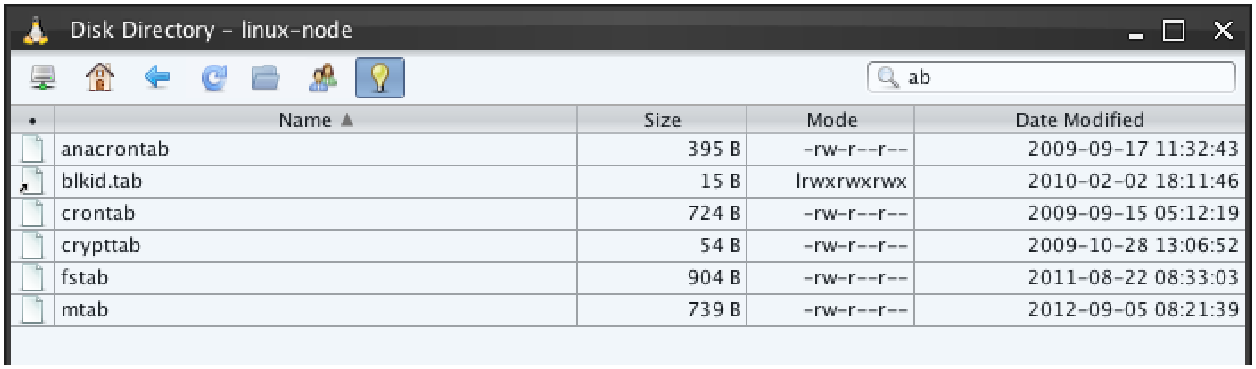
Items: 15, Dir: /tmp/ Offline

Fig. 6.18: **Disk Directory Window - Offline Browsing**



•	Name ▲	Size	Mode	Date Modified
	Unavailable		----------	

Fig. 6.19: **Disk Directory Window - Offline Browsing**



•	Name ▲	Size	Mode	Date Modified
🗋	anacrontab	395 B	-rw-r--r--	2009-09-17 11:32:43
↗🗋	blkid.tab	15 B	lrwxrwxrwx	2010-02-02 18:11:46
🗋	crontab	724 B	-rw-r--r--	2009-09-15 05:12:19
🗋	crypttab	54 B	-rw-r--r--	2009-10-28 13:06:52
🗋	fstab	904 B	-rw-r--r--	2011-08-22 08:33:03
🗋	mtab	739 B	-rw-r--r--	2012-09-05 08:21:39

Fig. 6.20: **Directory Window Search**

The following icons are used to represent Links in the Disk Directory window:

Symbol	Meaning
↗🗋	Link pointing to a file
↗🗀	Link is file, pointing to a directory
↗🗂	Link is directory, pointing to a directory

If the Link points to a directory, the operator may navigate to the destination by double-clicking the entry. If the Link is a directory ↗🗂, the Disk Directory window will simply navigate to the directory. However, if the Link is a file ↗🗀, the Disk Directory window will navigate to the specified target directory.

To view the target path of a Link, see *File Information*.

File Information

Detailed file or directory information can be viewed by right clicking on a file in the directory listing and selecting the "Get Info" menu item.

Selecting the "Get Info" menu item in the drop down menu displays a separate dialog with detailed information about the file or directory. The dialog shows the file name, file size in bytes, mode, UID, GID, access time, creation time, and modification time. If the selected file is a symbolic link the full target link path is displayed.

Note: In some cases, the actual target path of a Link may be relative to the location of the Link itself. If the target file or directory exists, the relative path will be converted to an absolute path.

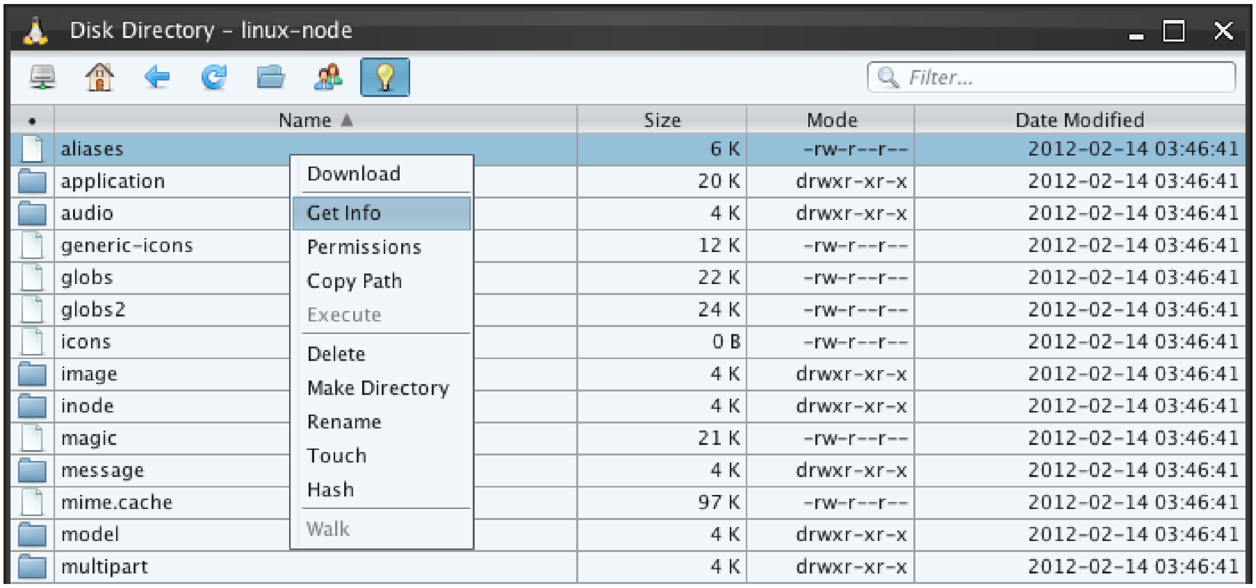
Copy Path

Allows an operator to copy the full path of the selected file in the Directory Window to the clipboard. This is done by right clicking a file in the directory listing and selecting "Copy Path".

A log entry is added for each path that is copied in the Logs window displaying the full path.

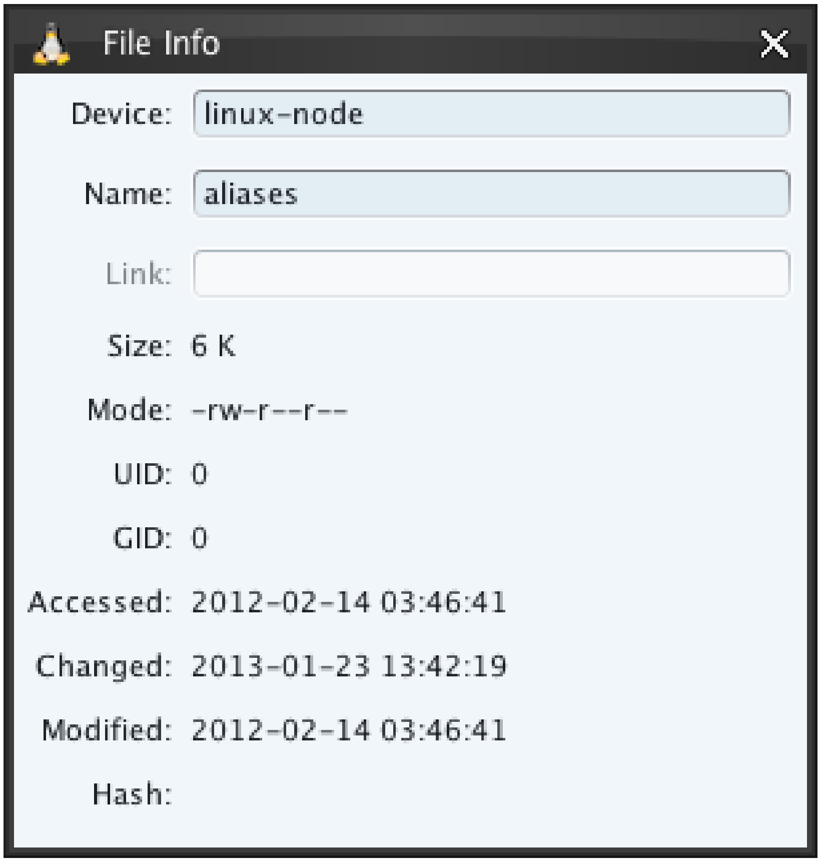
Execute

The "Execute" menu item is a shortcut to the Shell Module's execute capability. See *Shell Execute*.



•	Name ▲	Size	Mode	Date Modified
	aliases	6 K	-rw-r--r--	2012-02-14 03:46:41
	application	20 K	drwxr-xr-x	2012-02-14 03:46:41
	audio	4 K	drwxr-xr-x	2012-02-14 03:46:41
	generic-icons	12 K	-rw-r--r--	2012-02-14 03:46:41
	globs	22 K	-rw-r--r--	2012-02-14 03:46:41
	globs2	24 K	-rw-r--r--	2012-02-14 03:46:41
	icons	0 B	-rw-r--r--	2012-02-14 03:46:41
	image	4 K	drwxr-xr-x	2012-02-14 03:46:41
	inode	4 K	drwxr-xr-x	2012-02-14 03:46:41
	magic	21 K	-rw-r--r--	2012-02-14 03:46:41
	message	4 K	drwxr-xr-x	2012-02-14 03:46:41
	mime.cache	97 K	-rw-r--r--	2012-02-14 03:46:41
	model	4 K	drwxr-xr-x	2012-02-14 03:46:41
	multipart	4 K	drwxr-xr-x	2012-02-14 03:46:41

Fig. 6.21: **Directory Menu - Get Info Option**



Device: linux-node

Name: aliases

Link:

Size: 6 K

Mode: -rw-r--r--

UID: 0

GID: 0

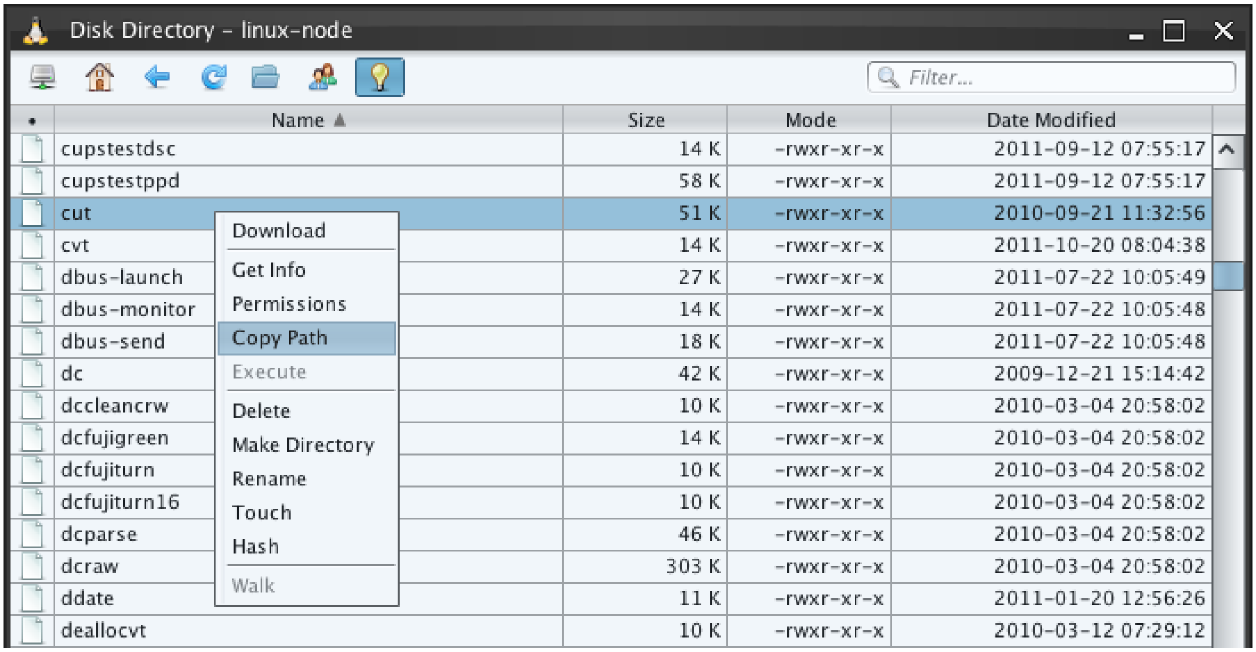
Accessed: 2012-02-14 03:46:41

Changed: 2013-01-23 13:42:19

Modified: 2012-02-14 03:46:41

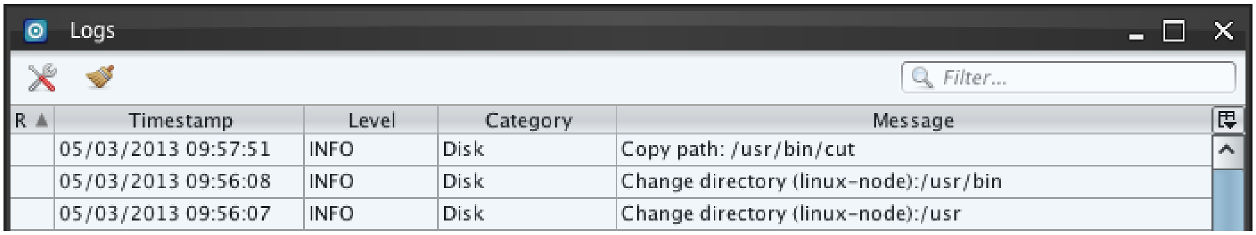
Hash:

Fig. 6.22: **File Information Dialog**



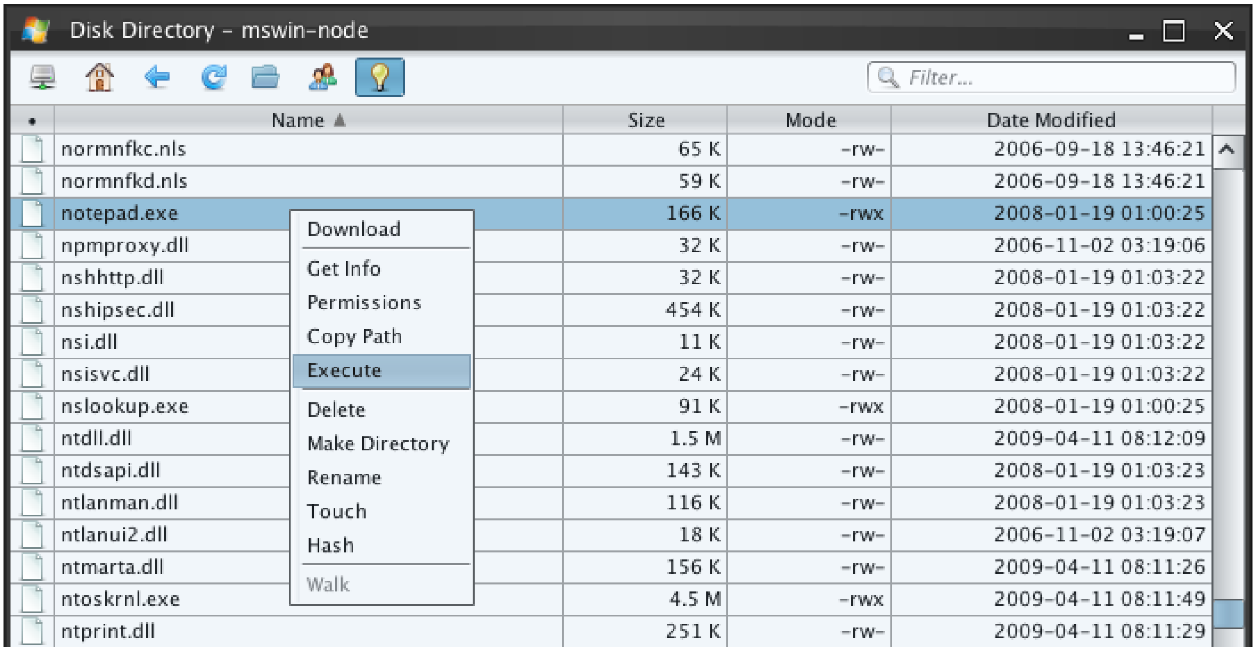
•	Name ▲	Size	Mode	Date Modified
	cupstestdsc	14 K	-rwxr-xr-x	2011-09-12 07:55:17
	cupstestppd	58 K	-rwxr-xr-x	2011-09-12 07:55:17
	cut	51 K	-rwxr-xr-x	2010-09-21 11:32:56
	cvt	14 K	-rwxr-xr-x	2011-10-20 08:04:38
	dbus-launch	27 K	-rwxr-xr-x	2011-07-22 10:05:49
	dbus-monitor	14 K	-rwxr-xr-x	2011-07-22 10:05:48
	dbus-send	18 K	-rwxr-xr-x	2011-07-22 10:05:48
	dc	42 K	-rwxr-xr-x	2009-12-21 15:14:42
	dccleancrw	10 K	-rwxr-xr-x	2010-03-04 20:58:02
	dcfujigreen	14 K	-rwxr-xr-x	2010-03-04 20:58:02
	dcfujiturn	10 K	-rwxr-xr-x	2010-03-04 20:58:02
	dcfujiturn16	10 K	-rwxr-xr-x	2010-03-04 20:58:02
	dcparse	46 K	-rwxr-xr-x	2010-03-04 20:58:02
	dcraw	303 K	-rwxr-xr-x	2010-03-04 20:58:02
	ddate	11 K	-rwxr-xr-x	2011-01-20 12:56:26
	deallocvt	10 K	-rwxr-xr-x	2010-03-12 07:29:12

Fig. 6.23: **Directory Menu - Copy Path**



R ▲	Timestamp	Level	Category	Message
	05/03/2013 09:57:51	INFO	Disk	Copy path: /usr/bin/cut
	05/03/2013 09:56:08	INFO	Disk	Change directory (linux-node):/usr/bin
	05/03/2013 09:56:07	INFO	Disk	Change directory (linux-node):/usr

Fig. 6.24: **Copy Path Log Entry**



•	Name ▲	Size	Mode	Date Modified
	normnfkc.nls	65 K	-rw-	2006-09-18 13:46:21
	normnfkd.nls	59 K	-rw-	2006-09-18 13:46:21
	notepad.exe	166 K	-rwx	2008-01-19 01:00:25
	npmproxy.dll	32 K	-rw-	2006-11-02 03:19:06
	nshhttp.dll	32 K	-rw-	2008-01-19 01:03:22
	nshipsec.dll	454 K	-rw-	2008-01-19 01:03:22
	nsi.dll	11 K	-rw-	2008-01-19 01:03:22
	nsisvc.dll	24 K	-rw-	2008-01-19 01:03:22
	nslookup.exe	91 K	-rwx	2008-01-19 01:00:25
	ntdll.dll	1.5 M	-rw-	2009-04-11 08:12:09
	ntdsapi.dll	143 K	-rw-	2008-01-19 01:03:23
	ntlanman.dll	116 K	-rw-	2008-01-19 01:03:23
	ntlanui2.dll	18 K	-rw-	2006-11-02 03:19:07
	ntmarta.dll	156 K	-rw-	2009-04-11 08:11:26
	ntoskrnl.exe	4.5 M	-rwx	2009-04-11 08:11:49
	ntprint.dll	251 K	-rw-	2009-04-11 08:11:29

Fig. 6.25: **Directory Menu - Execute**

Note: The Shell module must be loaded on the Local Control, remote Node, and the file must have the execute 'x' flag set.

Delete

Files and empty directories can be deleted from Nodes and Controls by right clicking on the item in the directory listing and selecting the "Delete" menu item.

Deleted files are overwritten X number of times where X is the value set in the *Preferences* panel. A non-empty directory cannot be deleted even though the option is available in the right click menu. Multiple files and directories can be deleted using native multi select. Delete attempts can be viewed by refreshing the current directory listing. The files selected for deletion are shown in a Delete dialog when the Delete option is selected in the right click menu. This dialog provides a final review of the files selected. Clicking the "OK" button will delete the files, while clicking the "Cancel" button will abort the delete operation.

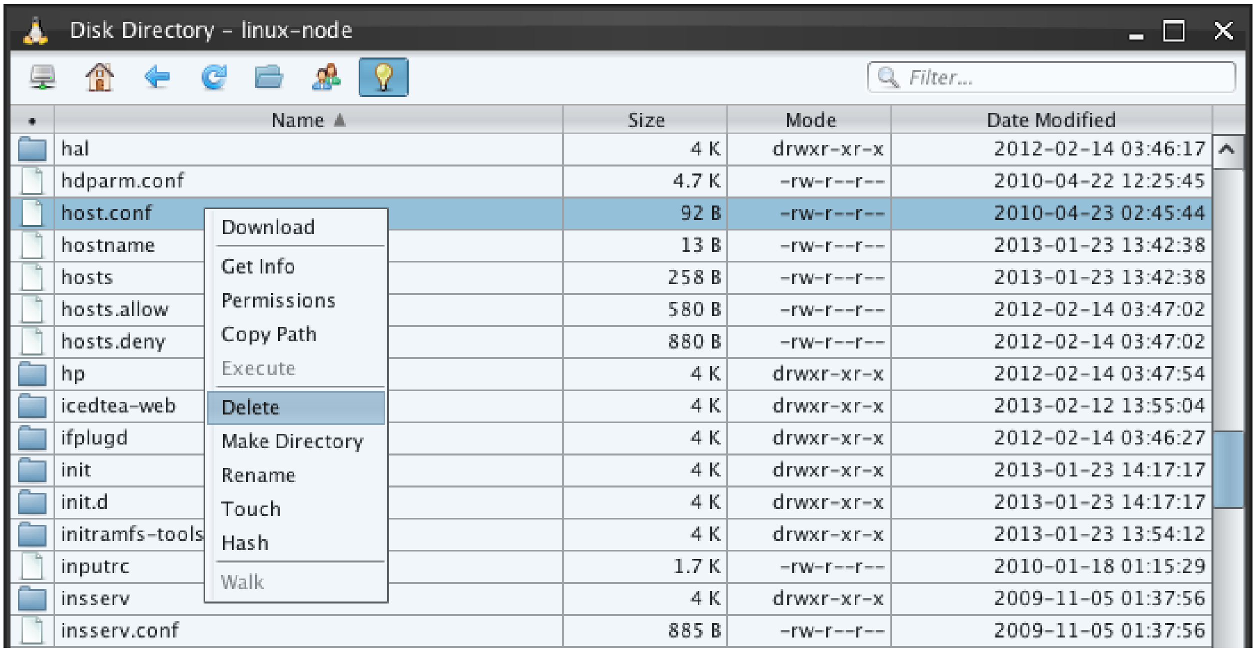
Note: The status of a delete operation is recorded (success/error) in the log window upon completion of the action. A progress bar is also shown in the Status Bar of the Directory Window for the duration of the operation.

Rename

Directory entries can be renamed by right clicking on the item in the directory listing and selecting the "Rename" menu item. This operation is only allowed during online browsing.

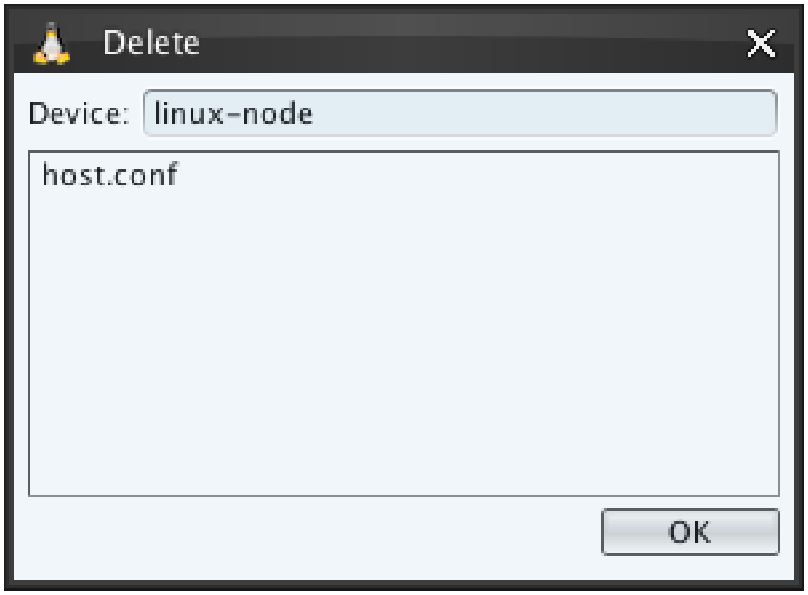
An entry is renamed to what the operator provides in the dialog. The "Refresh" button must be pressed in online browsing mode to confirm the operation was successful. This operation may fail silently if the name provided is invalid, the name conflicts with an existing entry in the current directory, or the file-system permissions are read-only.

Note: The status of a rename operation is recorded (success/error) in the log window upon completion of the action. A progress bar is also shown in the Status Bar of the Directory Window for the duration of the operation.



•	Name ▲	Size	Mode	Date Modified
	hal	4 K	drwxr-xr-x	2012-02-14 03:46:17
	hdparm.conf	4.7 K	-rw-r--r--	2010-04-22 12:25:45
	host.conf	92 B	-rw-r--r--	2010-04-23 02:45:44
	hostname	13 B	-rw-r--r--	2013-01-23 13:42:38
	hosts	258 B	-rw-r--r--	2013-01-23 13:42:38
	hosts.allow	580 B	-rw-r--r--	2012-02-14 03:47:02
	hosts.deny	880 B	-rw-r--r--	2012-02-14 03:47:02
	hp	4 K	drwxr-xr-x	2012-02-14 03:47:54
	icedtea-web	4 K	drwxr-xr-x	2013-02-12 13:55:04
	ifplugd	4 K	drwxr-xr-x	2012-02-14 03:46:27
	init	4 K	drwxr-xr-x	2013-01-23 14:17:17
	init.d	4 K	drwxr-xr-x	2013-01-23 14:17:17
	initramfs-tools	4 K	drwxr-xr-x	2013-01-23 13:54:12
	inputrc	1.7 K	-rw-r--r--	2010-01-18 01:15:29
	insserv	4 K	drwxr-xr-x	2009-11-05 01:37:56
	insserv.conf	885 B	-rw-r--r--	2009-11-05 01:37:56

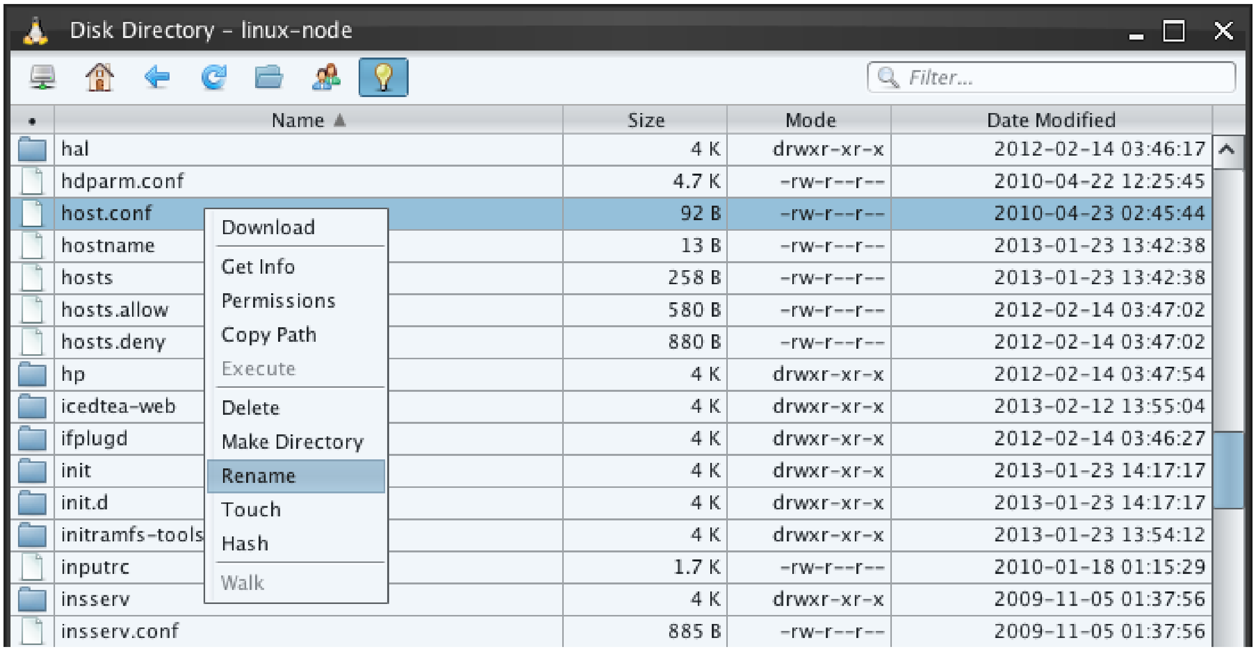
Fig. 6.26: **Directory Menu - Delete Option**



Device: linux-node

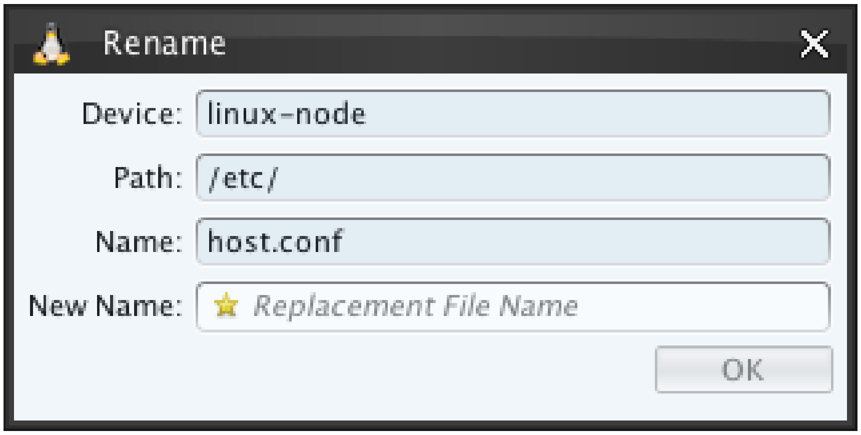
host.conf

Fig. 6.27: **Delete Dialog**



Name	Size	Mode	Date Modified
hal	4 K	drwxr-xr-x	2012-02-14 03:46:17
hdparm.conf	4.7 K	-rw-r--r--	2010-04-22 12:25:45
host.conf	92 B	-rw-r--r--	2010-04-23 02:45:44
hostname	13 B	-rw-r--r--	2013-01-23 13:42:38
hosts	258 B	-rw-r--r--	2013-01-23 13:42:38
hosts.allow	580 B	-rw-r--r--	2012-02-14 03:47:02
hosts.deny	880 B	-rw-r--r--	2012-02-14 03:47:02
hp	4 K	drwxr-xr-x	2012-02-14 03:47:54
icedtea-web	4 K	drwxr-xr-x	2013-02-12 13:55:04
ifplugd	4 K	drwxr-xr-x	2012-02-14 03:46:27
init	4 K	drwxr-xr-x	2013-01-23 14:17:17
init.d	4 K	drwxr-xr-x	2013-01-23 14:17:17
initramfs-tools	4 K	drwxr-xr-x	2013-01-23 13:54:12
inputrc	1.7 K	-rw-r--r--	2010-01-18 01:15:29
insserv	4 K	drwxr-xr-x	2009-11-05 01:37:56
insserv.conf	885 B	-rw-r--r--	2009-11-05 01:37:56

Fig. 6.28: **Directory Menu - Make Directory Option**



Device: linux-node

Path: /etc/

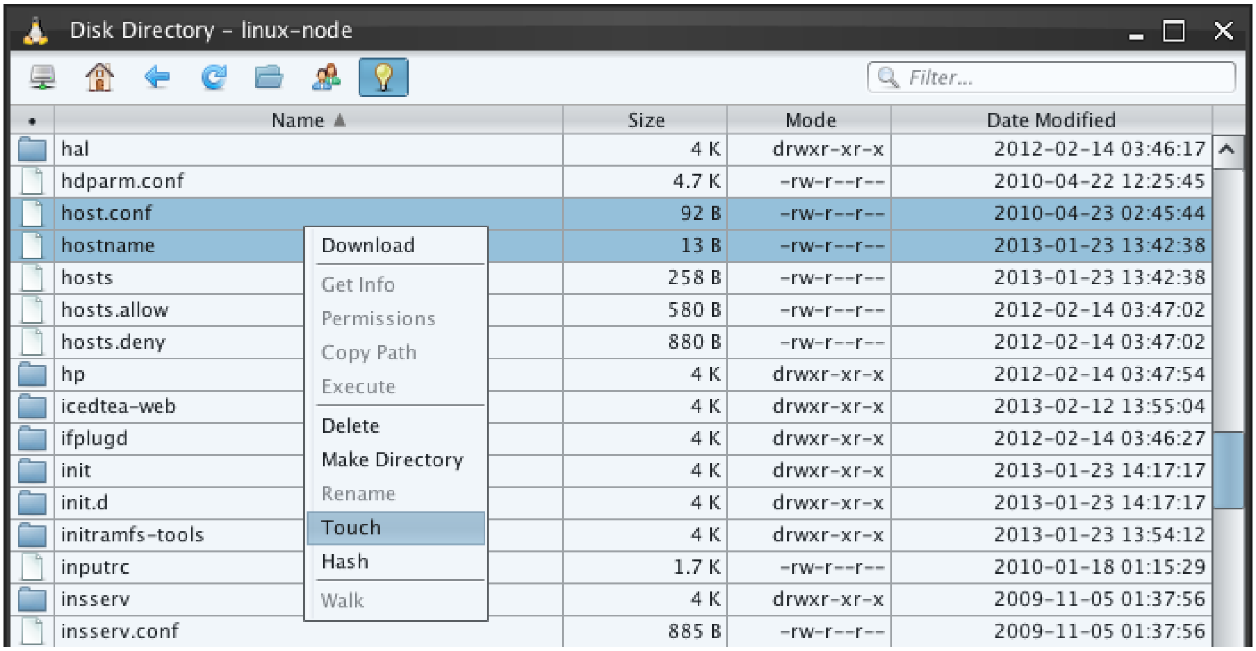
Name: host.conf

New Name: Replacement File Name

Fig. 6.29: **Rename Dialog**

Touch

Multiple directory entries can be touched by right clicking on the items in the directory listing and selecting the “Touch” menu item. This operation is only allowed during online browsing.



•	Name ▲	Size	Mode	Date Modified
	hal	4 K	drwxr-xr-x	2012-02-14 03:46:17
	hdparm.conf	4.7 K	-rw-r--r--	2010-04-22 12:25:45
	host.conf	92 B	-rw-r--r--	2010-04-23 02:45:44
	hostname	13 B	-rw-r--r--	2013-01-23 13:42:38
	hosts	258 B	-rw-r--r--	2013-01-23 13:42:38
	hosts.allow	580 B	-rw-r--r--	2012-02-14 03:47:02
	hosts.deny	880 B	-rw-r--r--	2012-02-14 03:47:02
	hp	4 K	drwxr-xr-x	2012-02-14 03:47:54
	icedtea-web	4 K	drwxr-xr-x	2013-02-12 13:55:04
	ifplugd	4 K	drwxr-xr-x	2012-02-14 03:46:27
	init	4 K	drwxr-xr-x	2013-01-23 14:17:17
	init.d	4 K	drwxr-xr-x	2013-01-23 14:17:17
	initramfs-tools	4 K	drwxr-xr-x	2013-01-23 13:54:12
	inputrc	1.7 K	-rw-r--r--	2010-01-18 01:15:29
	insserv	4 K	drwxr-xr-x	2009-11-05 01:37:56
	insserv.conf	885 B	-rw-r--r--	2009-11-05 01:37:56

Fig. 6.30: **Directory Menu - Touch Option**

The Timezone field indicates the timezone that will be used to process the times specified by the operator. Daylight Savings Time will always be turned off for the timezone. Thus, the times specified by the operator may appear different if Daylight Savings Time is enabled on the device. This field cannot be modified.

On Unix devices, only the Modified time may be set. On Windows devices, only the Created and Modified times may be set. The remaining fields are disabled and may be altered by the Operating System independent of the touch operation. The time format to enter is: yyyy-mm-dd hh:mm:ss. The minimum time that may be set is: 1970-01-01 00:00:00, and the maximum time that may be set is: 2038-01-19 03:14:07. Once the desired times are set, click “OK” to touch the files.

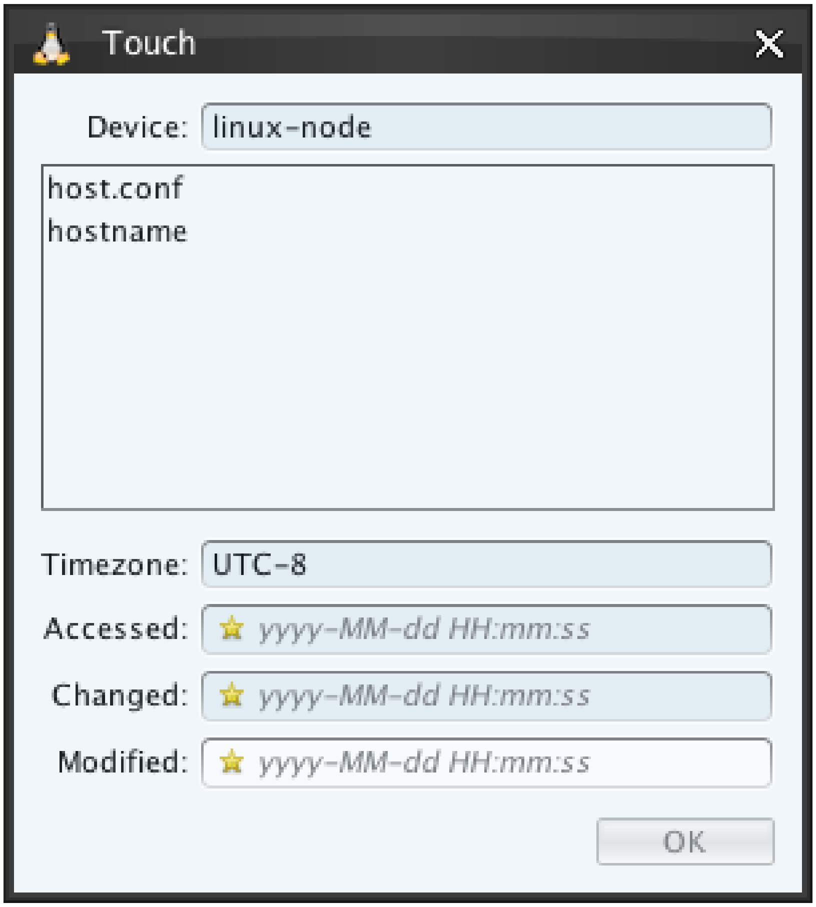
Hash

Files hashes (MD5 algorithm) can be fetched by selecting one or more files, right clicking on one of the selected files, and choosing the “Hash” menu item.

The results of this file operation are stored in the database, logged in the Log Window, and presented to an operator in the File Info Dialog when *File Information* is requested.

Make Directory

New empty directories can be made from Nodes and Controls by right clicking on any entry in the directory listing and selecting the “Make Directory” menu item. This operation is only allowed during online browsing. When adding a new directory to a currently empty path right click on the “Empty” placeholder.



Device: linux-node

host.conf
hostname

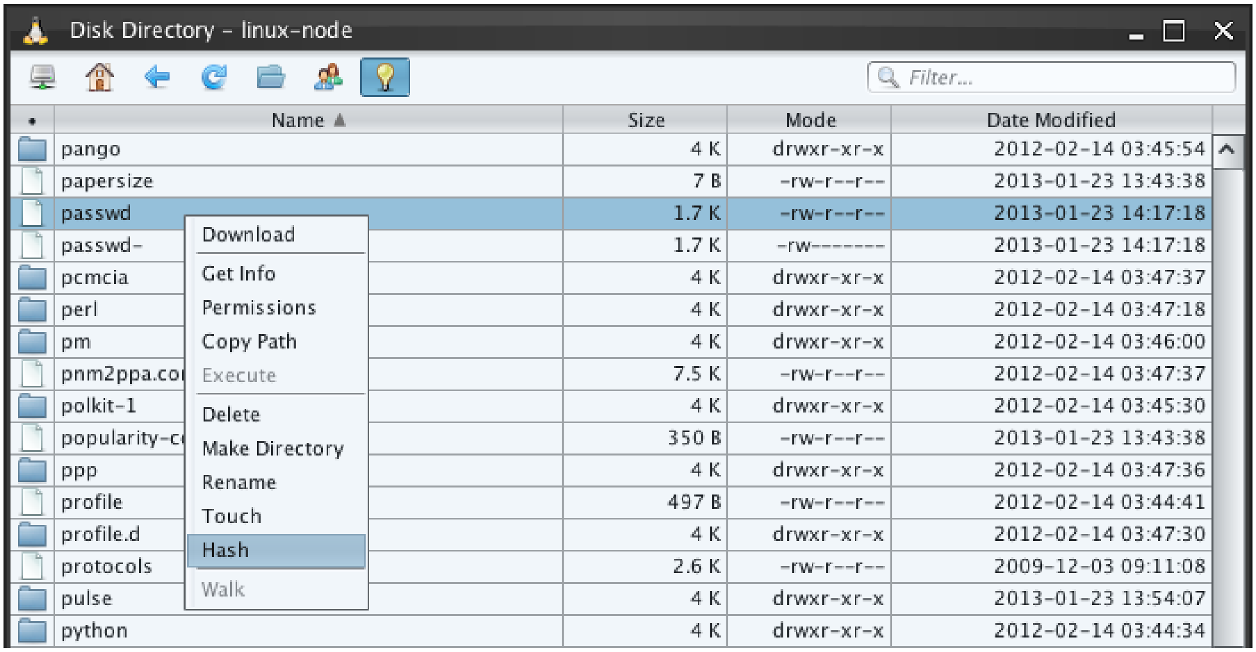
Timezone: UTC-8

Accessed: yyyy-MM-dd HH:mm:ss

Changed: yyyy-MM-dd HH:mm:ss

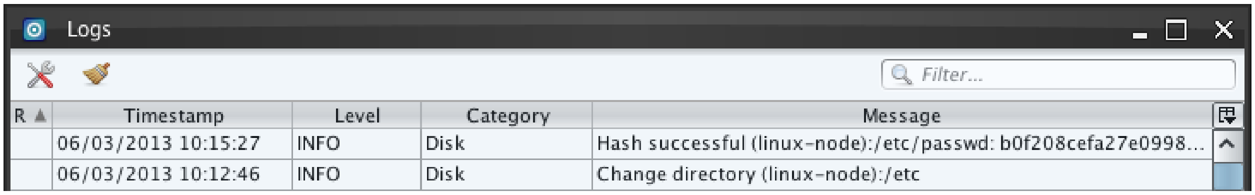
Modified: yyyy-MM-dd HH:mm:ss

Fig. 6.31: **Touch Dialog**



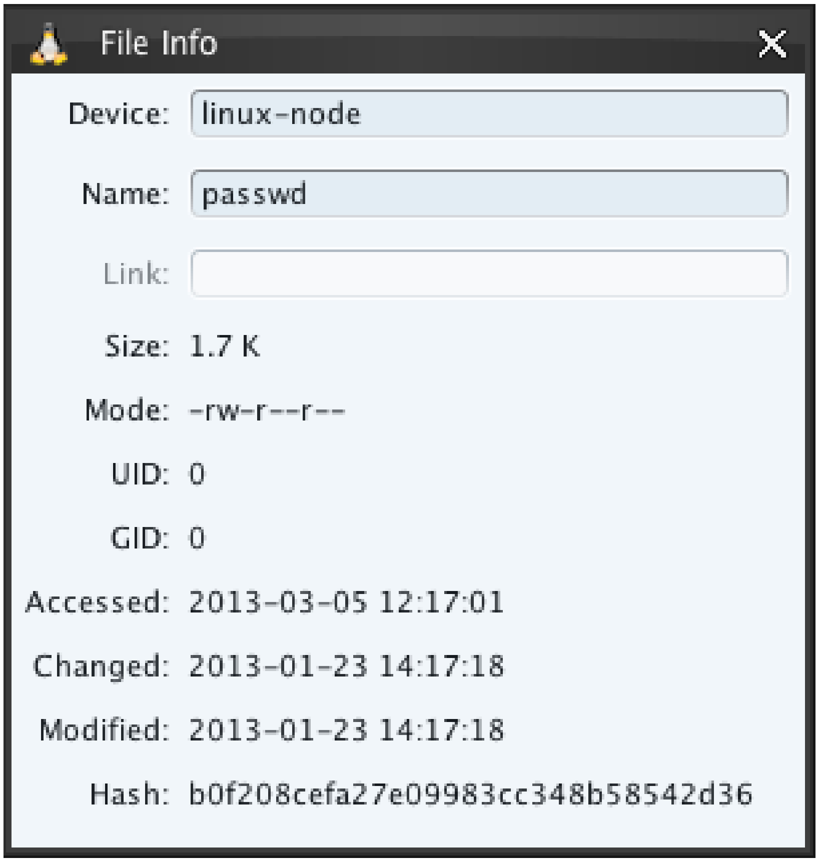
•	Name ▲	Size	Mode	Date Modified
	pango	4 K	drwxr-xr-x	2012-02-14 03:45:54
	papersize	7 B	-rw-r--r--	2013-01-23 13:43:38
	passwd	1.7 K	-rw-r--r--	2013-01-23 14:17:18
	passwd-	1.7 K	-rw-------	2013-01-23 14:17:18
	pcmcia	4 K	drwxr-xr-x	2012-02-14 03:47:37
	perl	4 K	drwxr-xr-x	2012-02-14 03:47:18
	pm	4 K	drwxr-xr-x	2012-02-14 03:46:00
	pnm2ppa.co	7.5 K	-rw-r--r--	2012-02-14 03:47:37
	polkit-1	4 K	drwxr-xr-x	2012-02-14 03:45:30
	popularity-c	350 B	-rw-r--r--	2013-01-23 13:43:38
	ppp	4 K	drwxr-xr-x	2012-02-14 03:47:36
	profile	497 B	-rw-r--r--	2012-02-14 03:44:41
	profile.d	4 K	drwxr-xr-x	2012-02-14 03:47:30
	protocols	2.6 K	-rw-r--r--	2009-12-03 09:11:08
	pulse	4 K	drwxr-xr-x	2013-01-23 13:54:07
	python	4 K	drwxr-xr-x	2012-02-14 03:44:34

Fig. 6.32: **Directory Menu - Hash Option**



R ▲	Timestamp	Level	Category	Message
	06/03/2013 10:15:27	INFO	Disk	Hash successful (linux-node):/etc/passwd: b0f208cefa27e0998...
	06/03/2013 10:12:46	INFO	Disk	Change directory (linux-node):/etc

Fig. 6.33: **Log Window - Hash Entry**



Device: linux-node

Name: passwd

Link:

Size: 1.7 K

Mode: -rw-r--r--

UID: 0

GID: 0

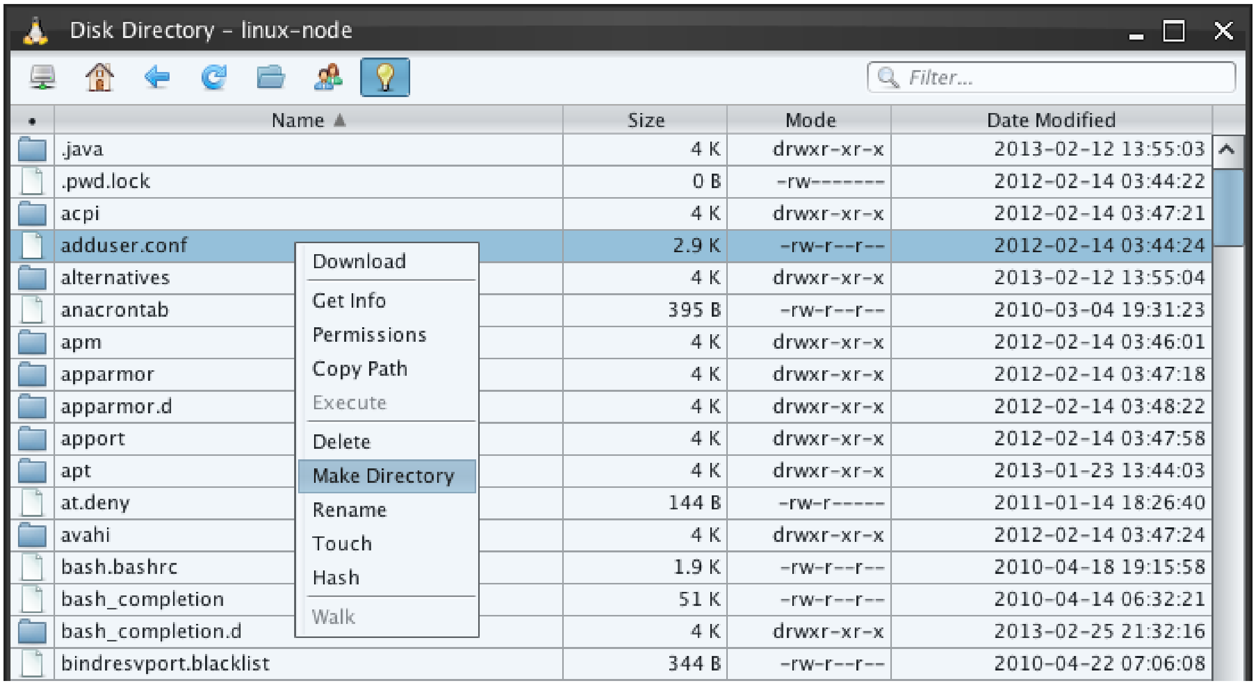
Accessed: 2013-03-05 12:17:01

Changed: 2013-01-23 14:17:18

Modified: 2013-01-23 14:17:18

Hash: b0f208cefa27e09983cc348b58542d36

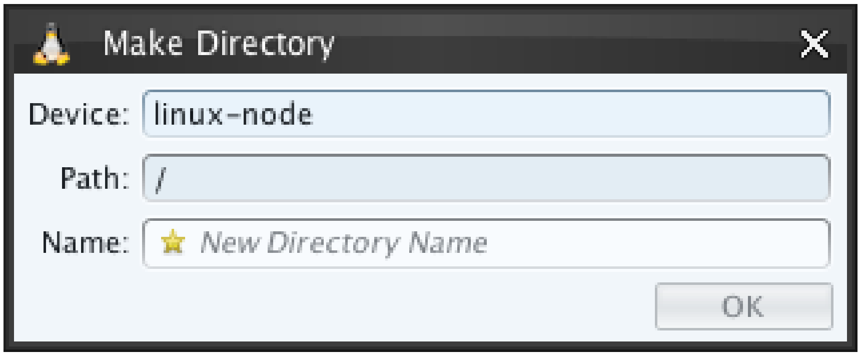
Fig. 6.34: **File Info Dialog with Hash**



•	Name ▲	Size	Mode	Date Modified
	.java	4 K	drwxr-xr-x	2013-02-12 13:55:03
	.pwd.lock	0 B	-rw-------	2012-02-14 03:44:22
	acpi	4 K	drwxr-xr-x	2012-02-14 03:47:21
	adduser.conf	2.9 K	-rw-r--r--	2012-02-14 03:44:24
	alternatives	4 K	drwxr-xr-x	2013-02-12 13:55:04
	anacrontab	395 B	-rw-r--r--	2010-03-04 19:31:23
	apm	4 K	drwxr-xr-x	2012-02-14 03:46:01
	apparmor	4 K	drwxr-xr-x	2012-02-14 03:47:18
	apparmor.d	4 K	drwxr-xr-x	2012-02-14 03:48:22
	apport	4 K	drwxr-xr-x	2012-02-14 03:47:58
	apt	4 K	drwxr-xr-x	2013-01-23 13:44:03
	at.deny	144 B	-rw-r-----	2011-01-14 18:26:40
	avahi	4 K	drwxr-xr-x	2012-02-14 03:47:24
	bash.bashrc	1.9 K	-rw-r--r--	2010-04-18 19:15:58
	bash_completion	51 K	-rw-r--r--	2010-04-14 06:32:21
	bash_completion.d	4 K	drwxr-xr-x	2013-02-25 21:32:16
	bindresvport.blacklist	344 B	-rw-r--r--	2010-04-22 07:06:08

Fig. 6.35: **Directory Menu - Make Directory Option**

A new directory is created with the operator's provided name entered in the dialog. The "Refresh" button must be pressed in online browsing mode to confirm the directory's creation. This operation may fail silently if the name provided is invalid, the name conflicts with an existing entry in the current directory, or the file-system permissions are read-only.



Device: linux-node

Path: /

Name: New Directory Name

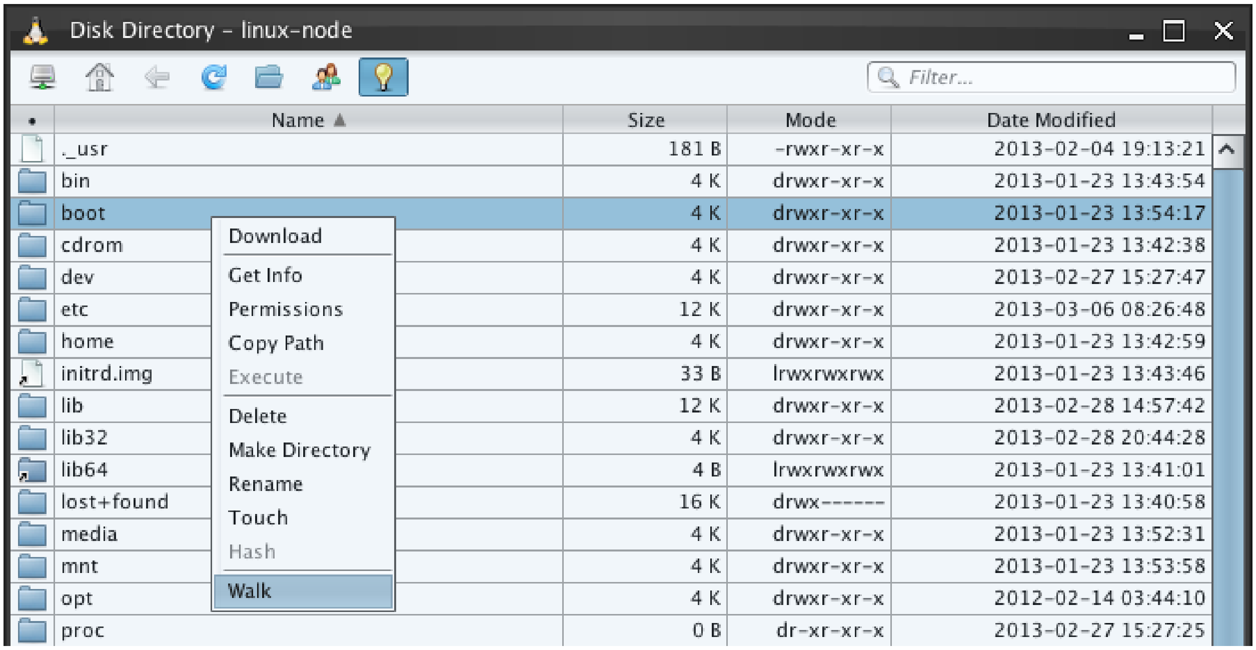
Fig. 6.36: **Make Directory Dialog**

Note: The new directories are created in the current directory of the Display Window and NOT as a subdirectory of a selected folder in the current directory's listing.

Note: The status of a make directory operation is recorded (success/error) in the log window upon completion of the action. A progress bar is also shown in the Status Bar of the Directory Window for the duration of the operation.

Walk

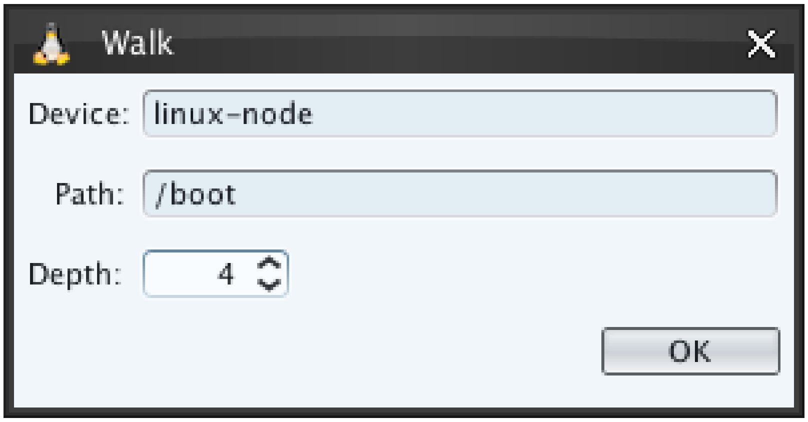
Subdirectories can be walked for later offline browsing by right clicking the subdirectory's folder in the Directory Listing and selecting the "Walk" menu item. A new walk request is sent to the node. Once the Node acknowledges the request is added to the Directory Walk table in the Disk Window. The Walk menu item is only enabled for directories.



•	Name ▲	Size	Mode	Date Modified
	._usr	181 B	-rwxr-xr-x	2013-02-04 19:13:21
	bin	4 K	drwxr-xr-x	2013-01-23 13:43:54
	boot	4 K	drwxr-xr-x	2013-01-23 13:54:17
	cdrom	4 K	drwxr-xr-x	2013-01-23 13:42:38
	dev	4 K	drwxr-xr-x	2013-02-27 15:27:47
	etc	12 K	drwxr-xr-x	2013-03-06 08:26:48
	home	4 K	drwxr-xr-x	2013-01-23 13:42:59
	initrd.img	33 B	lrwxrwxrwx	2013-01-23 13:43:46
	lib	12 K	drwxr-xr-x	2013-02-28 14:57:42
	lib32	4 K	drwxr-xr-x	2013-02-28 20:44:28
	lib64	4 B	lrwxrwxrwx	2013-01-23 13:41:01
	lost+found	16 K	drwx------	2013-01-23 13:40:58
	media	4 K	drwxr-xr-x	2013-01-23 13:52:31
	mnt	4 K	drwxr-xr-x	2013-01-23 13:53:58
	opt	4 K	drwxr-xr-x	2012-02-14 03:44:10
	proc	0 B	dr-xr-xr-x	2013-02-27 15:27:25

Fig. 6.37: **Directory Menu - Walk**

The Directory Window's Walk Dialog is nearly identical to the Directory Walk panel's dialog (*Start Directory Walk*). This walk starts from the specified path the operator clicked with the "Walk" menu item.



Device: linux-node

Path: /boot

Depth: 4

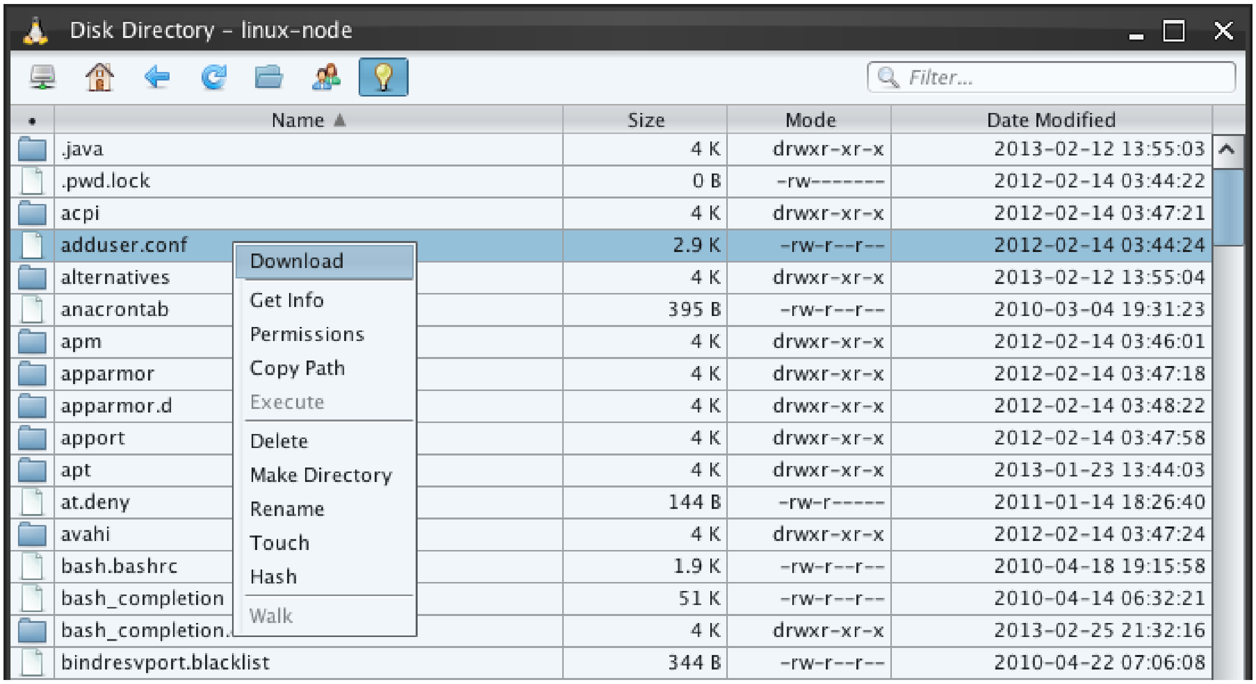
Fig. 6.38: **Directory Menu - Walk**

6.3 Transfers

Files can be transferred from the Local Control to a Node, from a Node to the Local Control, or from a Node to another Node. Files are transferred using a delta-encoding algorithm. This allows an operator to start a file transfer where they left off if the connection was interrupted or quickly sync up files without having to transfer the entire file to/from a Node. At the completion of all file transfers an MD5 checksum is performed and compared to the original file to ensure the file transferred successfully. If a file already exists with the same name as the one being transferred it will be overwritten.

6.3.1 Download

The Download option provides a shortcut to transfer files from a Node or Control back to the Local Control. The Local Directory text field in the Disk Preferences Panel displays the path where files will be saved when they are downloaded. This path is set during a sync of the Local Control and Horizon. If an operator changes this path it will only affect new downloads. To initiate a download an operator must select one or more files in a Directory Listing Table and choose the Download option in the right click menu.



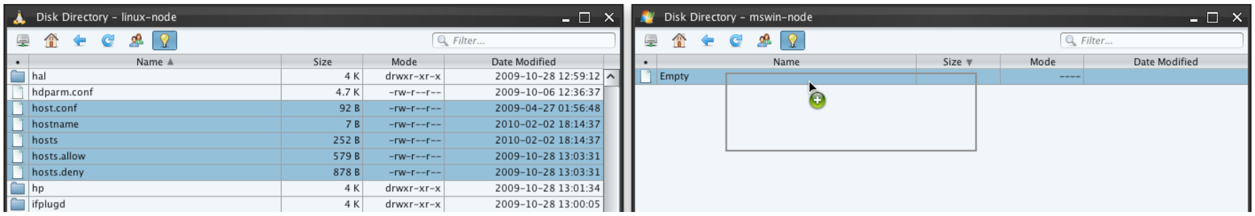
•	Name ▲	Size	Mode	Date Modified
	.java	4 K	drwxr-xr-x	2013-02-12 13:55:03
	.pwd.lock	0 B	-rw-------	2012-02-14 03:44:22
	acpi	4 K	drwxr-xr-x	2012-02-14 03:47:21
	adduser.conf	2.9 K	-rw-r--r--	2012-02-14 03:44:24
	alternatives	4 K	drwxr-xr-x	2013-02-12 13:55:04
	anacrontab	395 B	-rw-r--r--	2010-03-04 19:31:23
	apm	4 K	drwxr-xr-x	2012-02-14 03:46:01
	apparmor	4 K	drwxr-xr-x	2012-02-14 03:47:18
	apparmor.d	4 K	drwxr-xr-x	2012-02-14 03:48:22
	apport	4 K	drwxr-xr-x	2012-02-14 03:47:58
	apt	4 K	drwxr-xr-x	2013-01-23 13:44:03
	at.deny	144 B	-rw-r-----	2011-01-14 18:26:40
	avahi	4 K	drwxr-xr-x	2012-02-14 03:47:24
	bash.bashrc	1.9 K	-rw-r--r--	2010-04-18 19:15:58
	bash_completion	51 K	-rw-r--r--	2010-04-14 06:32:21
	bash_completion.	4 K	drwxr-xr-x	2013-02-25 21:32:16
	bindresvport.blacklist	344 B	-rw-r--r--	2010-04-22 07:06:08

Fig. 6.39: **Directory Menu - Download Option**

Note: The Download option is not available in the right click menu of the Local Control's Disk Directory window.

6.3.2 Drag and Drop

To transfer files to/from a Node or Control a minimum of two Disk Directory windows must be open and actively connected. An operator must select one or more files in a Disk Directory Listing Table and drag them to another Disk Directory Listing Table. During the drag operation a plus sign will appear in the receiving table indicating the files can be dropped. Once the files have been dropped the file transfer will begin. The current directory in the receiving table is where the files will be stored if the file transfer is successful.



•	Name ▲	Size	Mode	Date Modified
📁	hal	4 K	drwxr-xr-x	2009-10-28 12:59:12
📄	hdparm.conf	4.7 K	-rw-r--r--	2009-10-06 12:36:37
📄	host.conf	92 B	-rw-r--r--	2009-04-27 01:56:48
📄	hostname	7 B	-rw-r--r--	2010-02-02 18:14:37
📄	hosts	252 B	-rw-r--r--	2010-02-02 18:14:37
📄	hosts.allow	579 B	-rw-r--r--	2009-10-28 13:03:31
📄	hosts.deny	878 B	-rw-r--r--	2009-10-28 13:03:31
📁	hp	4 K	drwxr-xr-x	2009-10-28 13:01:34
📁	ifplugd	4 K	drwxr-xr-x	2009-10-28 13:00:05

•	Name	Size ▼	Mode	Date Modified
📄	Empty		----	

Fig. 6.40: **Drag and Drop File Transfers**

Note: If a file with the same name exists in a directory on the Local Control the new file will be incremented to prevent files from being overwritten. The format of the new file name is as follows: <filename>.<#>.

6.3.3 Directory Transfer

A directory can be transferred through one of the following actions:

- Select the directory in a Directory Window, right click it, and select the "Download" option in the menu or
- Drag and drop the selected directory into another open Directory Window.

The actions above will bring up a Directory Transfer Dialog to allow the operator to review the files in the selected directory. Files are fetched from the database to avoid recursively searching through a directory on a Node and causing unnecessary network congestion when returning the results. The Directory Transfer Dialog provides a tree view of all of the files to be transferred along with basic stats below the tree. At the top of the dialog are two fields: Device and To/From. These fields change depending on whether a Download (Device - Source and To - Destination) was selected or a Drag and Drop (Device - Destination and From - Source) operation was performed.

- Stats Overview

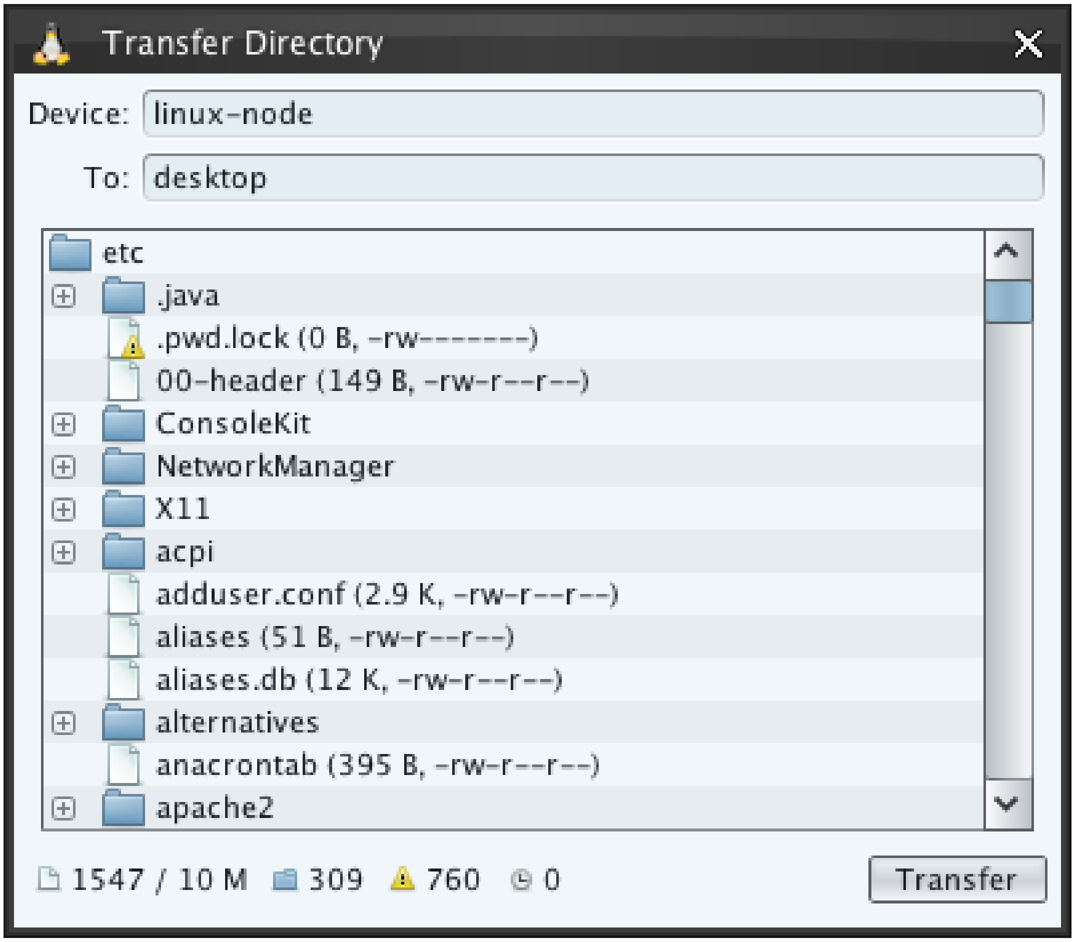
Symbol	Meaning
📄	Toal number of files that will be transferred / Total size (Mb) of files
📁	Total number of directories
⚠	Total number of files that cannot be transferred
🕒	Total number of files that are outdated

All files in the tree view include the following information: name, file size, and mode. Directories will only display their name and a '+' if they contain one or more files. Files that are not considered regular (mode does not begin with '-') or are of size 0 are marked with the following icon ⚠. Files that are outdated are marked with the following icon 🕒 to indicate the last time seen exceeded the value set in the *Preferences*.

If the file count exceeds the value set in *Preferences* a Stats Only view will be displayed. This is to prevent directories that contain a large number of files from using up memory in Horizon if its not necessary for the operator to review every file.

Directories that are either empty or have not been read/walked will return with zero files and will display 'Directory is Empty'.

After reviewing all of the files listed in the tree view and the stats an operator can click the Transfer button to begin transferring the files. If any of the Warning limits set in the *Preferences* are exceeded a final Warning Dialog will be displayed. This allows an operator one final chance to back out of the transfer.



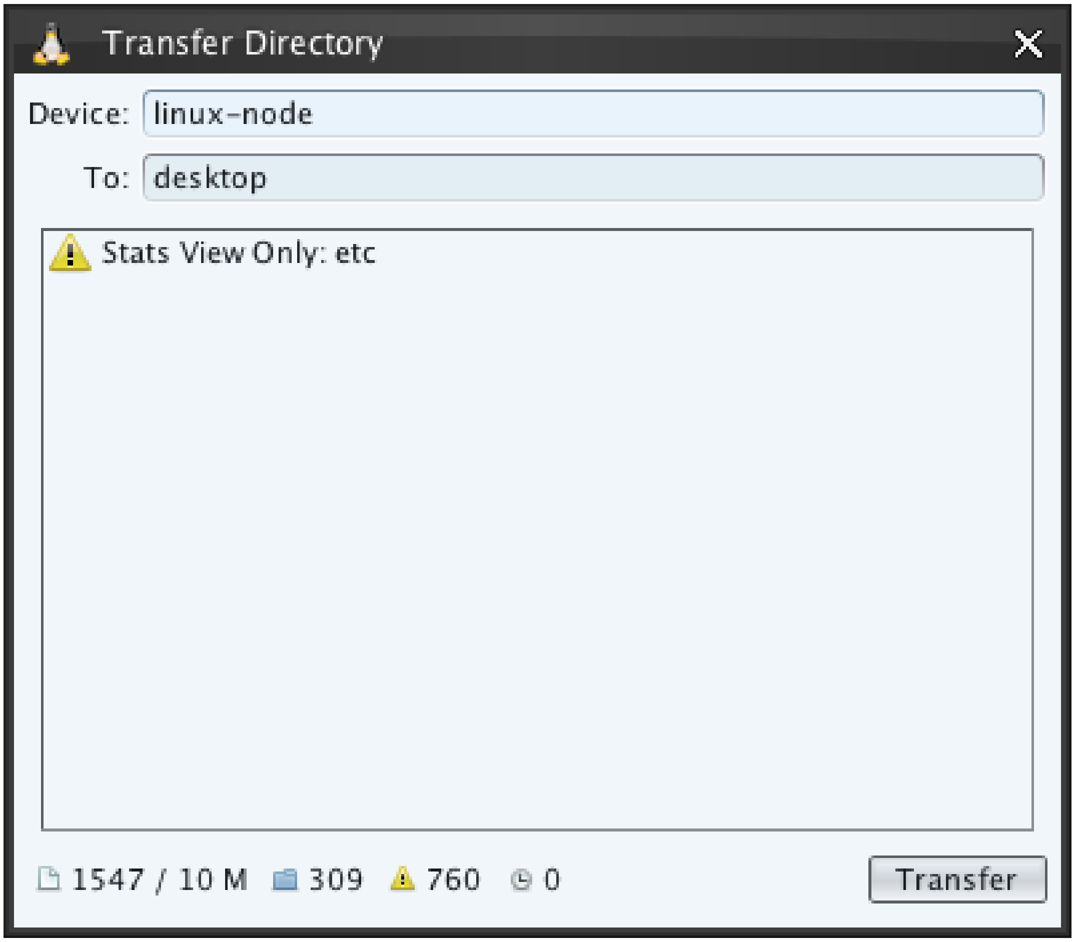
Device: linux-node

To: desktop

```
etc
  .java
  .pwd.lock (0 B, -rw-------)
  00-header (149 B, -rw-r--r--)
  ConsoleKit
  NetworkManager
  X11
  acpi
  adduser.conf (2.9 K, -rw-r--r--)
  aliases (51 B, -rw-r--r--)
  aliases.db (12 K, -rw-r--r--)
  alternatives
  anacrontab (395 B, -rw-r--r--)
  apache2
```

1547 / 10 M 309 ⚠ 760 0

Fig. 6.41: **Directory Transfer Dialog**



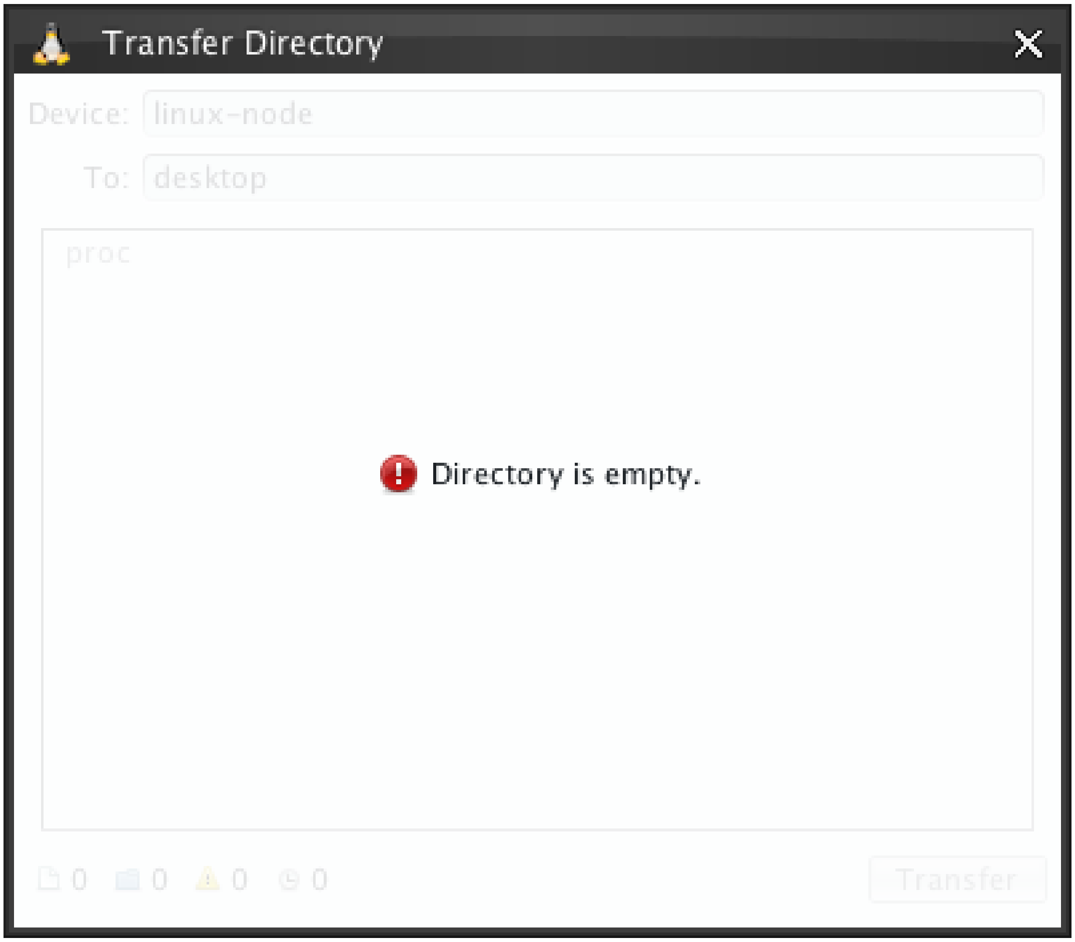
Device: linux-node

To: desktop

Stats View Only: etc

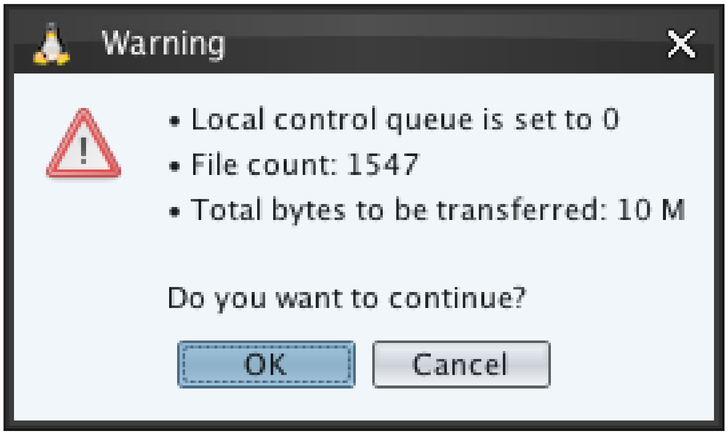
1547 / 10 M 309 760 0

Fig. 6.42: **Directory Transfer Dialog - Stats Only**



Directory is empty.

Fig. 6.43: **Directory Transfer Dialog - Empty Directory**



- Local control queue is set to 0
- File count: 1547
- Total bytes to be transferred: 10 M

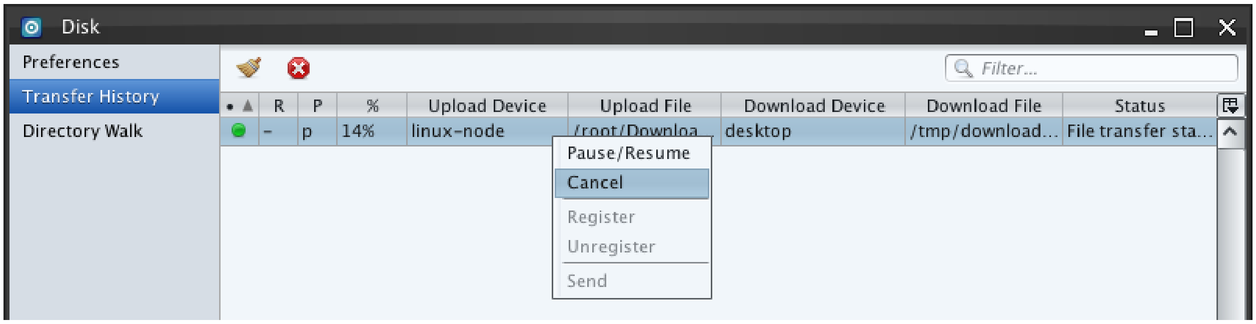
Do you want to continue?

Fig. 6.44: **Directory Transfer Warning Dialog**

> **Warning:** This feature is extremely powerful and can easily be abused if too many files are selected and transferred at one time. If the Local Control Queue is set to 0, all file transfer requests are sent and initiated on the source node potentially using up all resources on the given node.

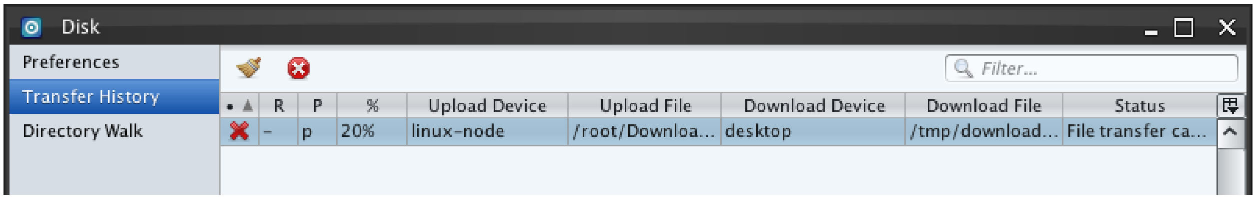
6.3.4 Cancel

Active file transfers in the Transfer History table can be cancelled by selecting the desired file transfer and choosing the Cancel option in the right click menu.



• ▲	R	P	%	Upload Device	Upload File	Download Device	Download File	Status
●	–	p	14%	linux-node	/root/Downloa	desktop	/tmp/download...	File transfer sta...

Fig. 6.45: **Transfer History Menu - Cancel Option**

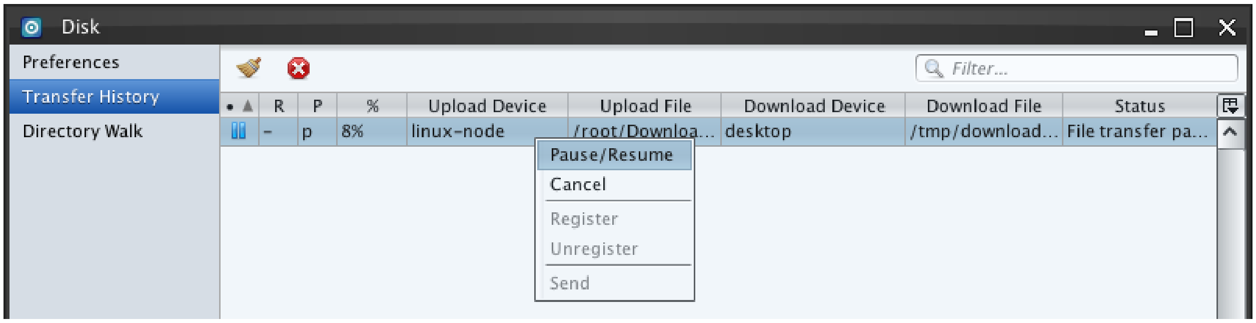


• ▲	R	P	%	Upload Device	Upload File	Download Device	Download File	Status
✗	–	p	20%	linux-node	/root/Downloa...	desktop	/tmp/download...	File transfer ca...

Fig. 6.46: **Cancelled Transfer**

6.3.5 Pause

Active file transfers in the Transfer History table can be paused by selecting the desired file transfer and choosing the Pause/Resume option in the right click menu.

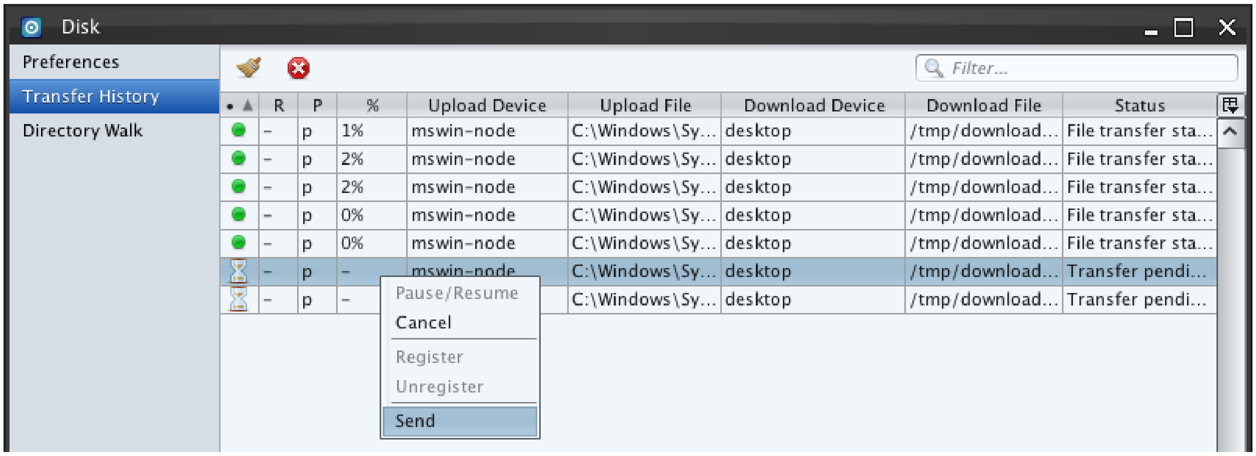


• ▲	R	P	%	Upload Device	Upload File	Download Device	Download File	Status
▮▮	–	p	8%	linux-node	/root/Downloa...	desktop	/tmp/download...	File transfer pa...

Fig. 6.47: **Paused Transfer**

6.3.6 Force Send

Queued file transfers on the Local Control can be forced up to the Upload Node by choosing the Send option in the right click menu.

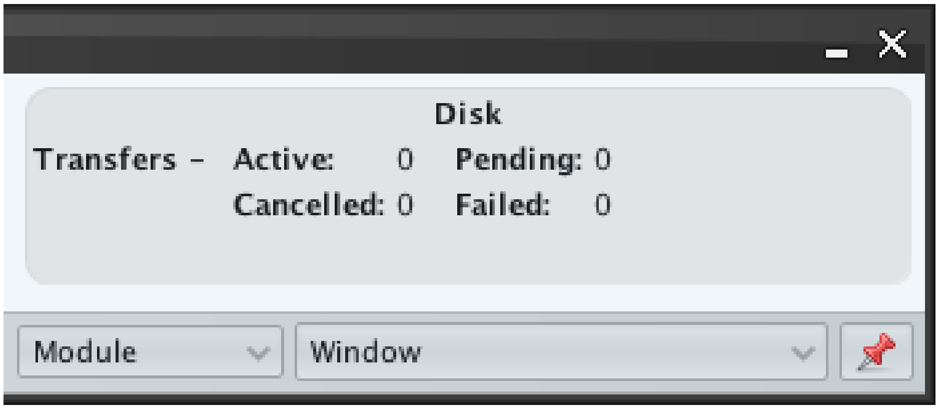


• ▲	R	P	%	Upload Device	Upload File	Download Device	Download File	Status
●	–	p	1%	mswin-node	C:\Windows\Sy...	desktop	/tmp/download...	File transfer sta...
●	–	p	2%	mswin-node	C:\Windows\Sy...	desktop	/tmp/download...	File transfer sta...
●	–	p	2%	mswin-node	C:\Windows\Sy...	desktop	/tmp/download...	File transfer sta...
●	–	p	0%	mswin-node	C:\Windows\Sy...	desktop	/tmp/download...	File transfer sta...
●	–	p	0%	mswin-node	C:\Windows\Sy...	desktop	/tmp/download...	File transfer sta...
⌛	–	p	–	mswin-node	C:\Windows\Sy...	desktop	/tmp/download...	Transfer pendi...
⌛	–	p	–		C:\Windows\Sy...	desktop	/tmp/download...	Transfer pendi...

Fig. 6.48: **Transfer History Menu - Send Option**

6.4 Dashboard Module Status

The Disk Module Status Panel in the Dashboard provides an operator with the number of active, pending, cancelled, and failed file transfers.



Disk

Transfers – Active: 0 Pending: 0

Cancelled: 0 Failed: 0

Fig. 6.49: **Module Status**

6.5 Limitations

- 32-bit Windows node executed on Microsoft Windows 64-bit versions will be unable to see any files in System32 that do not also appear in SysWOW64.

Note: The Wow64 subsystem is a lightweight compatibility layer that has similar interfaces on all 64-bit versions of

Windows. Its primary purpose is to create a 32-bit environment that provides the interfaces required to allow 32-bit Windows applications to run unmodified in the 64-bit system (http://en.wikipedia.org/wiki/WoW64).

- The Date Modified column in Directory Listings will appear empty even after refreshing in online browsing mode if the node returns system time zero. The Date Modified is not read during directory walks. Date Modified timestamps can also be verified using the Shell Module.

SHELL MODULE

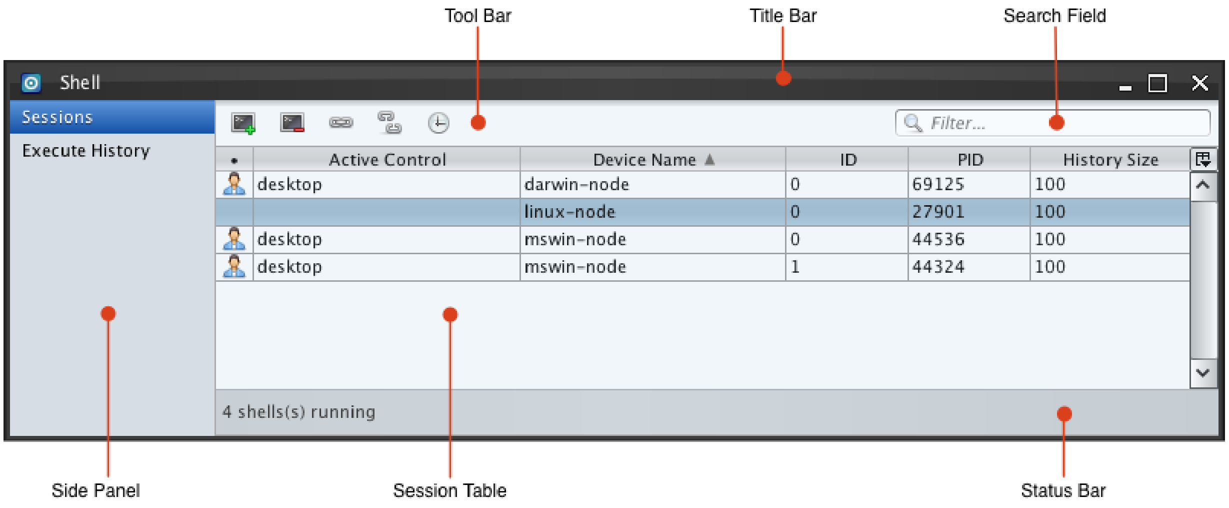
The Shell Module provides two features: 1) command line sessions that can be collaboratively managed by Controls and 2) a fully functional command line execution feature that provides a secondary means of executing binaries on remote Nodes. Sessions in the Shell Module are similar to remote terminal services like telnet or ssh; they provide a way to execute programs via a command line interface by using the device's shell. The Shell Module allows operators to create new sessions from any Control with the Shell Module activated. It allows the client's Local Control to attach to a remote session and interact with the Operating System terminal.

7.1 Shell Window

The Shell Window is a global window accessible from the Dashboard's "Module" combo box when the Local Control has activated the Shell Module. The Shell Window includes two panels, Sessions and *Execute History Panel*.

7.1.1 Sessions Panel

Opening the Shell Window and selecting the Sessions option in the side panel will display the Sessions panel:



•	Active Control	Device Name ▲	ID	PID	History Size
👤	desktop	darwin-node	0	69125	100
		linux-node	0	27901	100
👤	desktop	mswin-node	0	44536	100
👤	desktop	mswin-node	1	44324	100

4 shells(s) running

Fig. 7.1: **Shell Window - Sessions**

The Sessions panel includes a Tool Bar with all session management controls. The status bar provides feedback for pending session management interactions. The Session Table provides a listing of all sessions currently spawned in the mesh network. A Control is described as the *Active Control* when it is attached to a session. Only one Active Control can interact with a session at a time.

The Session Table is organized into six columns that describe the following:

- • (Status): The status column shows one of three icons describing the Active Control of the session.

Table 7.1: Session Status Icons

Symbol	Meaning
🧑	Locally attached session. The client's Local Control is attached to the session.
👤	Remotely attached session. Another Control is attached to the session.
No icon	The session exists, but no Control is attached.

- Active Control: This column displays the Device Name of the Active Control. It is blank when no Control is attached.
- Device Name: This column displays the Device Name of the remote Node the session was spawned on. It is never blank.
- ID: This column displays an instance identification number for the sessions. The identification number distinguishes between multiple sessions created on a particular device.
- PID: This column displays the Operating System's process identification number of the session.
- History Size: This column displays the byte size of the history buffer for each session.

7.1.2 Creating Sessions

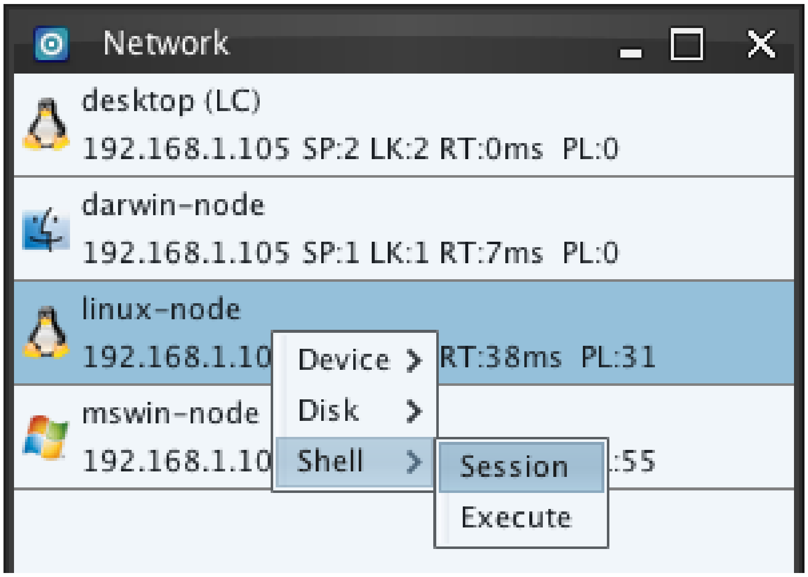
A new session can be created two ways:

- The first way is too right-click on the chosen device in Network Window and select "Session" from the Shell menu.
- The second way is to click the "Create Session" button in the Tool Bar.

This will launch the Create Session dialog. Select the target device from the combo box and click the "OK" button. This will create a new Session and add a new entry to the Session Table in the Shell Window.

The Control that creates a new session automatically becomes the Active Control of the session. The Shell Module provides a built-in Microsoft Windows shell with a limited Unix-like command set. The available built-in commands can be printed by typing *help*. Unix operating systems use the first shell that successfully launches from /bin or /usr/bin directories. The order attempted is always: /sbin/sh, bash, csh, ksh, sh. Users must verify the type of shell actually spawned themselves. /sbin/sh is attempted first for compatibility with Solaris 10.

Note: All Unix operating systems limit the number of shell sessions that can be spawned from a Node at any given



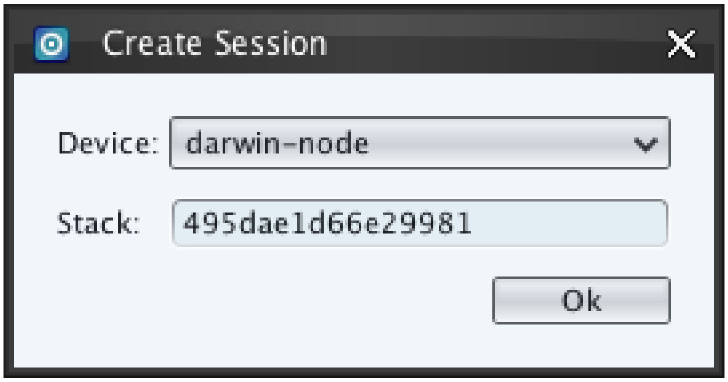
desktop (LC)
192.168.1.105 SP:2 LK:2 RT:0ms PL:0

darwin-node
192.168.1.105 SP:1 LK:1 RT:7ms PL:0

linux-node
192.168.1.10 RT:38ms PL:31

mswin-node
192.168.1.10 :55

Fig. 7.2: **Shell Popup Menu**



Device: darwin-node

Stack: 495dae1d66e29981

Fig. 7.3: **Create Session Dialog**

time. This limitation is usually 32 sessions. If the limit is breached no command prompt will appear in the new Session Window. Sessions can still be terminated using the button described in *Terminating Sessions*.

Warning: Each session on Microsoft Windows Nodes requires a new process to be spawned on the target system containing Fluxwire's built-in command shell for Windows. The secondary processes will all show the same process name as the Fluxwire Node executable in Task Manager or other process listing utilities.

7.1.3 Terminating Sessions

Sessions can be terminated in multiple ways.

- Selecting the session(s) in the Shell Window and pressing the Terminate Session button.
- Shutting down the device.
- Entering a built-in shell specific command at the command prompt i.e., *exit*.
- Supplying end of standard input notification to the shell such as pressing Control-D on most operating systems.

Warning: Closing the Session Window does not terminate the session; it detaches the session. Processes associated with the shell session remain active.

7.1.4 Attaching to a Session

An existing session can be attached to using the Shell Window.

Select the session(s) in the session table and press the attach button. Once attached, a new Session Window appears on the Active Control's clients. If the Local Control is already attached and has a Session Window, the existing Session Window is brought to front.

7.1.5 Detaching from a Session

An existing session can be detached from its Active Control in three ways.

- Selecting the session(s) in the Shell Window and pressing the Detach button.
- Closing the attached Session Window. See *Session Windows*.
- Another flx-gui client's Local Control causes a detach by pressing the Attach or Detach buttons on their Shell Window. See *Stealing a Session*.

  Warning: Operators are encouraged to detach sessions prior to exiting flx-gui; sessions can continue to send data to their Active Control if the session's executing program(s) produce new output.

7.1.6 Setting the Session History Buffer

A small cache of the most recent data sent from the session to the Active Control is saved. When a session is attached this buffer is sent to the new Active Control as a convenience to initially populate the Session Window with the last bit of information present before the session was last detached.

Select the session(s) of interest in the Session Table and click the History Buffer Button. A dialog box will appear to enter a new buffer size in bytes.

Note: Sometimes history buffers can cause display artifacts when first attaching to a session. This is due to misinterpreting the context of the history buffer for things like terminal escape sequences. Unix-based Operating Systems can usually issue a "reset" command to fix this.

7.1.7 Stealing a Session

Only one Control can be actively attached to a session at a time. It is possible to attach to a session that is already attached to by another Control. Under these circumstances the session is *stolen*. The Session Window on the former Active Control's clients will become disabled. The *Logs Window* will receive a message indicating the session was stolen and by which Control.

7.2 Session Windows

Session Windows provide interactive command-line terminals. These windows are created when sessions are attached to by Controls. A Session Windows' Title Bar identifies the device name and ID from the Shell Window's session table. The terminal scrollbar can be moved up and down to view review contents of the scrollback buffer. The Buffered Command Text Area at the bottom allows full commands to send to the remote terminal in one data message rather than a character at a time. Displays the Font Dialog, which allows selection fonts for the Session Terminal's contents and the font size. By default the font is monospaced (fixed-width) and 12 point. Changing the font or font size in one Session Window does not change the font in other existing Session Windows or new Session Windows. The Session Terminal always uses the "Plain" font styles, ignoring bold, italics, etc.

Note: Unicode content can produce unrenderable characters, such as squares or diamond-shaped question marks because the current font selected may not support received character codes. Use the Font Dialog to select another fixed-width font with better support for the particular language in use. Unicode characters are not supported on Microsoft Windows Nodes. See limitations for *Microsoft Windows*.

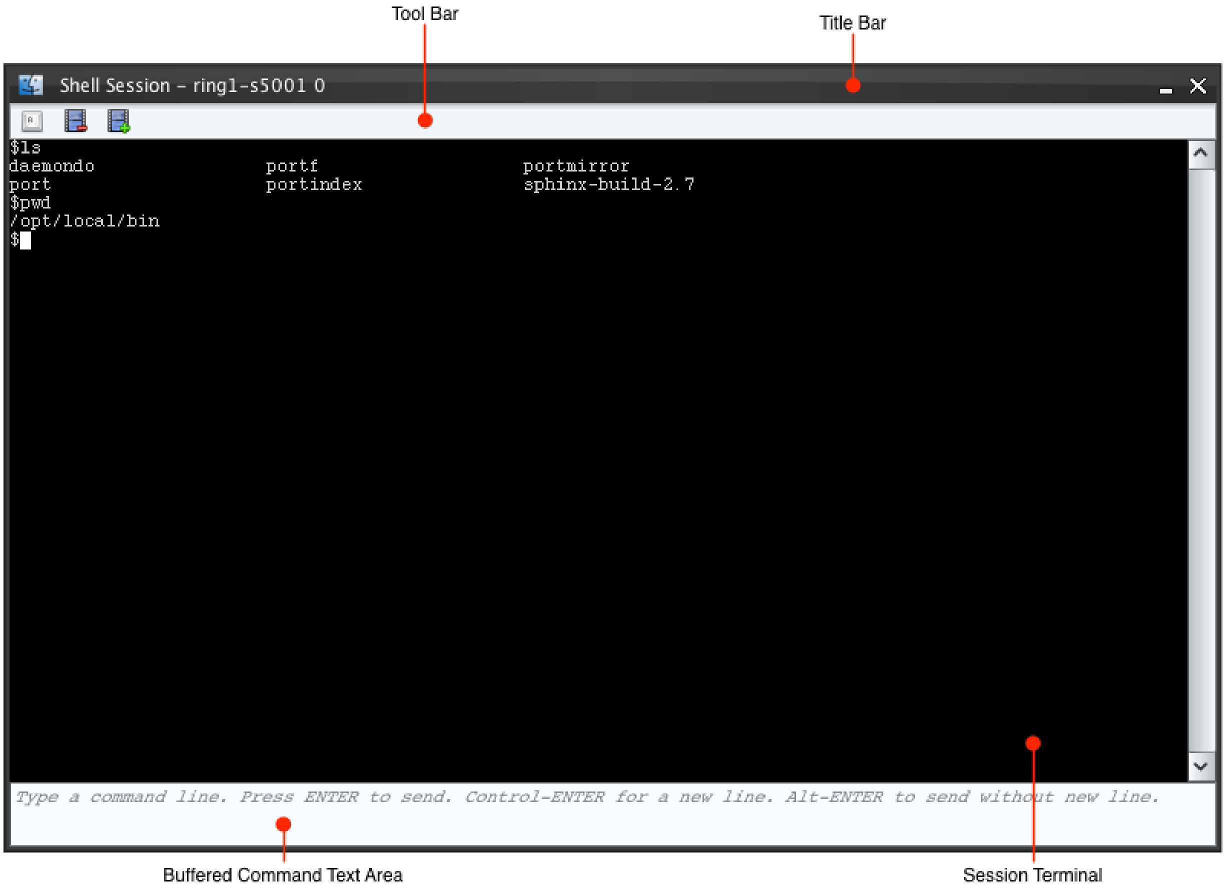
Increases the terminal's scrollback buffer by 100 lines. A larger scrollback buffer allows the user to scroll through more content using the scrollbar on the right-side of the window. Smaller scrollback buffers can make the terminal feel more responsive when the executing program is rapidly generating output to the terminal.

Decreases the terminal's rollback buffer by 100 lines. This button is not visible for sessions targeting Microsoft Windows.

Note: The scrollbar should be placed all the way down to see the command prompt.

When a session detaches from the Local Control while the window is open the terminal will be disabled, appearing gray, and the Title Bar receives an asterisk suffix. Once Session Windows are disabled they are permanently frozen. Their content remains on the screen for operator review. If the Local Control attaches back to the same session a new Session Window is created.

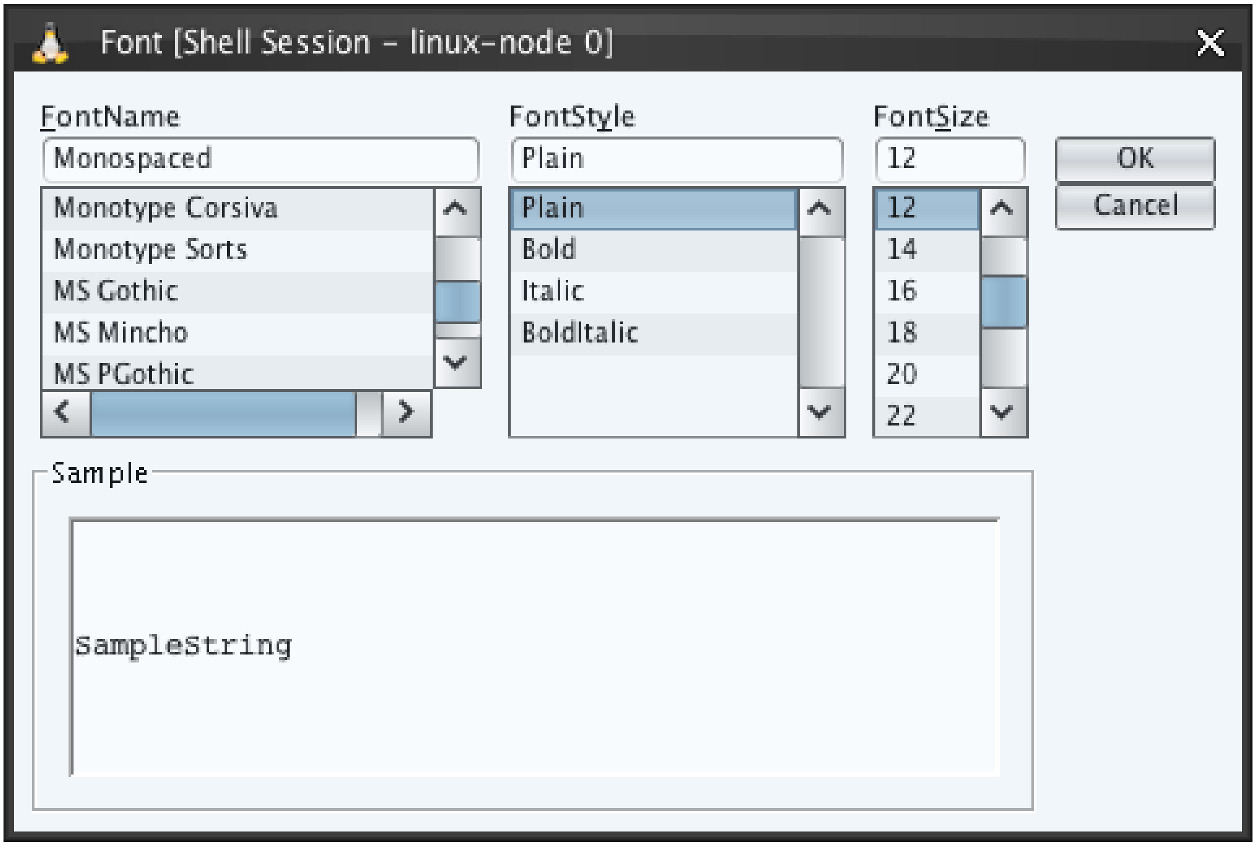
This toggle button only appears on Microsoft Windows devices. It is enabled by default. While enabled, backslashes typed in the Buffered Command Text Area will be automatically escaped when sent to the Terminal. For example: C:\Windows\System32 would become C:\\Windows\\System32 when sent. When directly typing in the terminal, automatic escaping does not occur.



```
$ls
daemondo                 portf                    portmirror
port                     portindex                sphinx-build-2.7
$pwd
/opt/local/bin
$
```

Type a command line. Press ENTER to send. Control-ENTER for a new line. Alt-ENTER to send without new line.

Fig. 7.4: **Session Window showing the Unix "top" Command**


Font [Shell Session - linux-node 0]

FontName: Monospaced

FontStyle: Plain

FontSize: 12

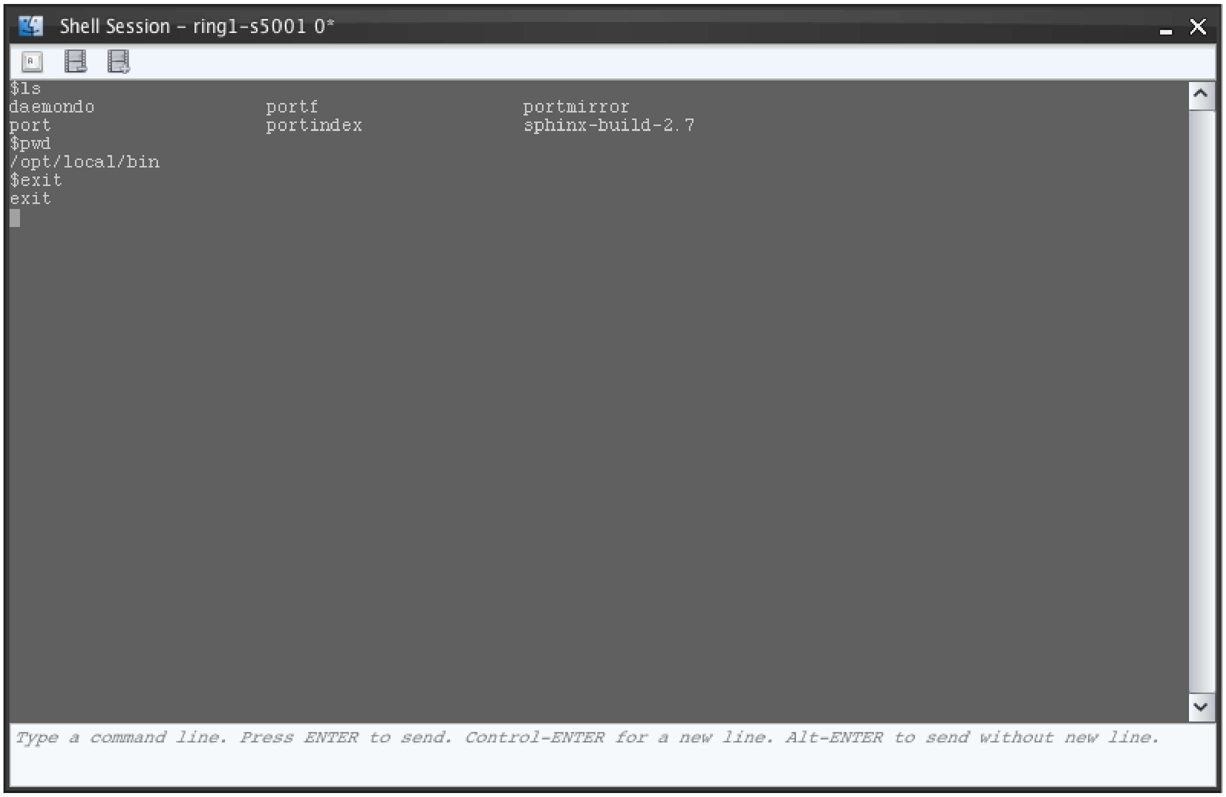
Sample

SampleString

7.2.1 Buffered Command Text Area

The text area at the bottom of Session Windows allow users to type full commands or multiple commands and then send the buffered command(s) in one network message to the remote node's terminal. The command entered is sent when the **Enter** key is pressed. When **Enter** is pressed the command along with a newline character is sent. If the user holds down the Alt key with **Enter** the newline is not appended. If the user holds down the Control key with **Enter** a new line is inserted into the buffered text area without immediately sending all of its contents. Several common keyboard shortcuts and a right click menu are also available in the text area.

- Cut: Control+X
- Copy: Control+C
- Paste: Control+V
- Undo: Control+Z
- Redo: Control+Shift+Z
- Focus Change: Tab. Changes focus to the terminal.
- Send with newline appended: ENTER



Shell Session – ring1-s5001 0*

```
$ls
daemondo                 portf                    portmirror
port                     portindex                sphinx-build-2.7
$pwd
/opt/local/bin
$exit
exit
```

Type a command line. Press ENTER to send. Control-ENTER for a new line. Alt-ENTER to send without new line.

Fig. 7.5: **Detached Session Window**

Table 7.2: Buffered Command Text Area Key Commands

Input	Meaning
Control+X	Cut
Control+C	Copy
Control+V	Paste
Control+Z	Undo
Control+Shift+Z	Redo
Tab	Changes the keyboard focus from the text area to the terminal.
Enter	Sends the buffered command **with** a newline appended.
Alt+Enter	Sends the buffered command **without** a newline appended.
Control+Enter	Adds a newline (enter keystroke) to the text area. Sends no message.

7.2.2 Copy and Pasting in a Session Window

Dragging the left mouse button across the text in the Session Terminal will highlight text. Click the Right-Mouse Button to bring up a pop-up menu and select “Copy” to copy the highlighted text. A “Paste” operation can be performed by clicking the Right-Mouse Button in the Session Terminal and selecting “Paste.”

Note: “Copy” operations will capture terminal output exactly as it appears in the Session Window. If the copy buffer spans across a line wrap a newline or other appropriate line wrapping characters are included in the copy buffer. Care should be taken when using this to copy a long command the wraps to multiple lines.

7.2.3 Job Killing (Ctrl-C) Support

On Unix Operating Systems, POSIX signals can be delivered through Ctrl-[Key] combinations. For example, Ctrl-C sends a SIGINT signal to the foreground process requesting execution to terminate. Microsoft Windows does not support POSIX signals. However, Session Windows emulate support for Ctrl-C *like* behavior. Ctrl-C can be used on all supported Operating Systems to kill processes running in the session.

Warning: Unix operating systems apply Ctrl-C only to the foreground process. It is possible for Microsoft Windows processes to detach or disassociate from the Node, which limits the effectiveness of emulated Ctrl-C support on Microsoft Windows.

7.2.4 Command History (Unix Operating Systems)

Warning: Most Unix shells maintain a command history log file. The Shell Module does not manipulate the target Node shell’s command history. Command history logging is specific to the target’s shell (bash, ksh, csh, etc). Command history logging must be manually disabled by users after creating each Shell Session. For example, bash command history can be disabled by issuing the command “unset HISTFILE” before issuing any other commands on the session.

7.2.5 Shell Session Log Files

Active Controls automatically log all shell sessions' activity in its base directory in the subfolder shell/session-logs. Log files for sessions currently open on Controls end in a random temporary file name and log files that have been finalized end in .log. The log file names contain the following information hyphen separated:

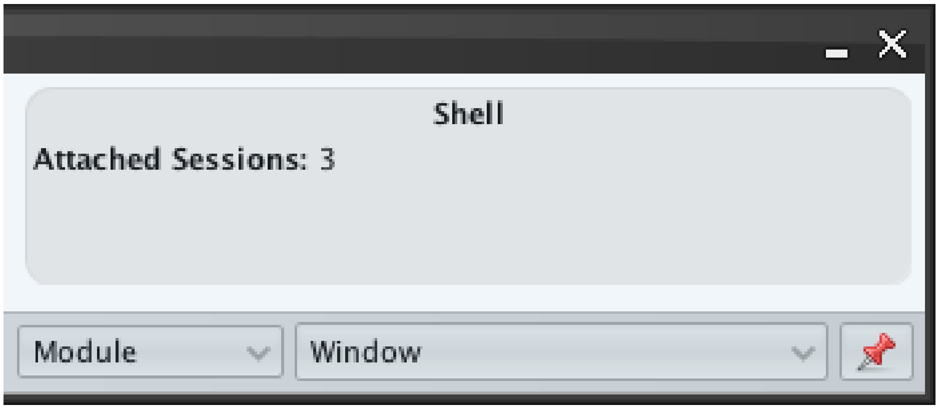
- Device Name of the Node as set in the Device Module
- Stack Address of the Node
- Shell session ID from the ID column of the Session table
- Attach count
- Date and time (finalized log files only)

The Attach Count starts at zero and increments each time a control attaches to the same shell session. When Controls share a session only the current Active Control performs logging. Consider the following example. Control A creates a session and detaches. Then Control B attaches to the session and detaches. Finally, Control A reattaches to the session and terminates it. This scenario results in two log files with attach counts 0 and 1 in Control A's base directory and one log file with an attach count of 0 in Control B's base directory.

Log file contents includes all terminal and other non-printable characters that the session outputs. Therefore, interactive commands (like *top*) that move the cursor, clear rows, and rewrites sections of the terminal cause a lot of terminal IO characters to be logged.

7.3 Dashboard Module Status

The Shell Module Status Panel in the Dashboard provides an operator with number of sessions attached to by the Local Control.



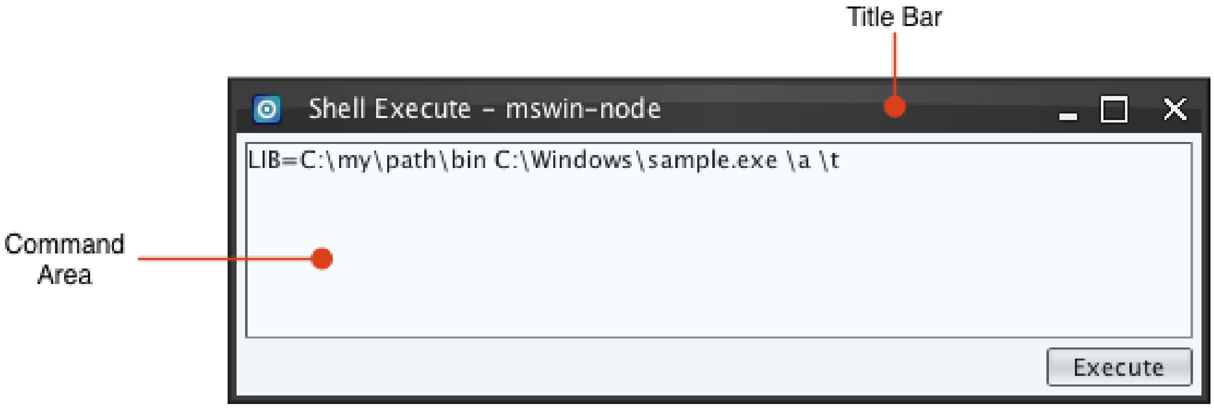
Attached Sessions: 3

Fig. 7.6: **Module Status**

7.4 Shell Execute

The Shell Module provides a fully functional command line feature that provides a secondary means of executing binaries on a remote Node. Binary path, arguments, and environment variables can be entered by an operator providing the necessary components to customize execution of remote binaries.

Right-clicking the selected Node in the Network Window and selecting “Execute” from the Shell menu brings up a Shell Execute Window.



```
LIB=C:\my\path\bin C:\Windows\sample.exe \a \t
```

Fig. 7.7: **Shell Execute Window**

Specify the command in the following order as shown in the figure above:

- Multiple environment variables such as PATH=/usr/local/bin HOME=/home/myhome
- Full path to the binary to execute
- Arguments to the executable.

Press “Execute” to parse the command and bring up the Execute Confirm Window.

After verifying the input, press “OK” to execute the command on the remote Node. The execution request is added to the Execution History Panel.

7.4.1 Execute History Panel

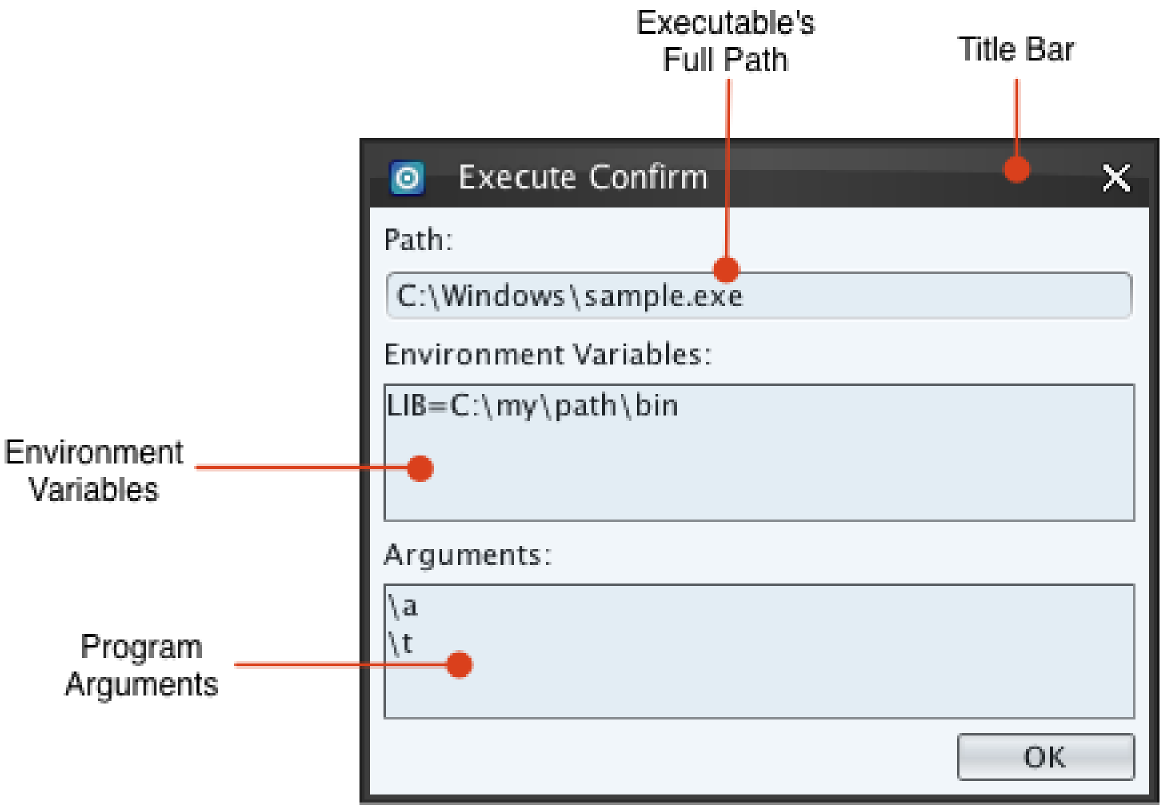
Opening the *Shell Window* and selecting the Execute History option in the side panel will display the Shell Execute feature’s history. Each time a command execution request is made an entry is added to the table.

The Execute History Table is organized into six columns that describe the following:

- • (State): This column is empty or displays an icon for the state of the command. ● if the command fails.

Table 7.3: Execute History Status Icons

Symbol	Meaning
●	The command failed to execute.
▮▮▮	The command has been sent and is waiting for a response.
No icon	Completed.



Path:

C:\Windows\sample.exe

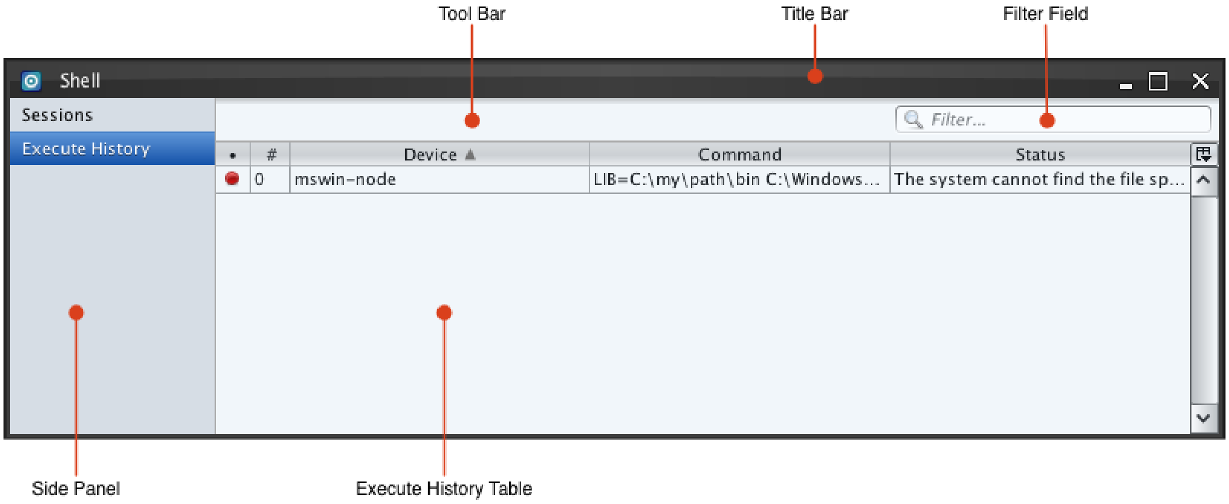
Environment Variables:

LIB=C:\my\path\bin

Arguments:

\a
\t

Fig. 7.8: **Shell Execute Confirm Window**



•	#	Device ▲	Command	Status
●	0	mswin-node	LIB=C:\my\path\bin C:\Windows...	The system cannot find the file sp...

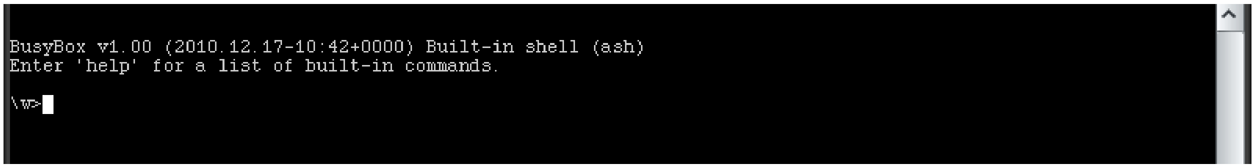
Fig. 7.9: **Shell Window - Execute History**

- Device: This column displays the Device Name of the remote Node the command was executed on.
- Command: This column displays the command that was executed.
- Status: This column displays error messages as to why the command failed, otherwise it is blank.

Note: Unix Operating Systems will not report failed execute attempts due limitations.

7.5 Limitations

1. On some on Unix operating systems the backspace key is not properly set to delete. Type the following command to set it: **stty erase <Backspace>**
2. A flx-gui client can abruptly detach a session from another Control by pressing the Attach or Detach buttons on their Shell Window. See *Stealing a Session*.
3. The Shell Module does not disable any command history logging by default. Operators must disable command history logging manually if desired. See *Command History (Unix Operating Systems)*.
4. Copy and pasting limitation. There is a known bug in some Linux distributions, including Fedora 10, where the Java Runtime Environment (JRE) fails to access the system clipboard correctly during Copy operations. The result of the bug is that a Copy operation performed in Horizon can be pasted inside any Horizon application window, but not other applications like gnome-terminal, gedit, etc. In other applications "Paste" menu items will appear disabled due to this JRE bug.
5. Copy and pasting limitation. When multiple lines are pasted into a Session Terminal while at a command prompt, the line endings can cause each line to be submitted like individual commands. This is because the copy operation copies what displays in Session Terminals. Session Terminals wrap command input onto multiple rows when it runs out of room on the prompt's row. If these multiple-row commands are copied, line endings are introduced in the command strings where the command was wrapped. If the copy buffer is pasted without removing these line endings (by using a text editor) commands can be misinterpreted.
6. Prompts can sometimes be interpreted incorrectly causing escape sequences to be displayed as shown below. This is more common on embedded systems that incorporate custom shells. This does not have any effect on the usability and functionality of the session but may be a distraction. The prompt on Unix systems can be fixed by exporting the PS1 environment variable with a new value (i.e., **export PS1="$"**).



```
BusyBox v1.00 (2010.12.17-10:42+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

\w>
```

Fig. 7.10: **Invalid Prompt on a x86 Mikrotik Node**

7.5.1 Microsoft Windows

1. On Microsoft Windows Ctrl-Z signals end of file instead of process suspension, whereas Unix operating systems commonly use Ctrl-D. The Shell Module's built-in Microsoft Windows shell emulates Unix style Ctrl-D end of file support.
2. The Shell Module's built-in Microsoft Windows shell emulates Unix like Ctrl-C process interruption behavior and attempts to terminate the current foreground process. It may be possible for Microsoft Windows processes to detach or disassociate from the Node, which prevents Fluxwire from successfully terminating the foreground process.

3. The Shell Module's built-in Microsoft Windows shell does not support Unicode at this time. The Session Window will display squares, diamonds, or question marks for these characters.
4. Jobs and background processes in the built-in Microsoft Windows' shells are not supported.
5. Executing shell scripts in the built-in Microsoft Windows' shell is an *experimental* feature.
6. The directory listing command, ls, will report "Permission denied" if a file is currently opened by another process with a flag set to deny shared access.
7. Some executables like schtasks.exe rely on Microsoft's Console API for interactive prompts and output. These executables will fail to process input and output correctly. For most utilities like schtasks.exe there are command line arguments that can alternatively be used to eliminate interactive prompts that cause this issue.
8. 64-bit Nodes that are spawned by a 32-bit application on a 64-bit version of Microsoft Windows may display the initial directory in a new shell session incorrectly. This only occurs if the 64-bit Node is within a directory in SysWOW64 that does not also exist in System32. This is due to the 32-bit application being forced to redirect access from System32 to SysWOW64.
9. Each session on Microsoft Windows Nodes requires a new process to be spawned on the target system containing the Shell Module's built-in Microsoft Windows shell. The secondary processes show the same process name as the Node's executable in Task Manager or other process listing utilities.
10. Microsoft Windows Nodes packed as a Dynamic Link Library (DLL) using the -l (library) option in flx-packer cannot spawn Shell Sessions. The Horizon GUI disables the "Session" menu item for these Nodes.

CHAPTER

EIGHT

CHAT MODULE

The Chat module provides the ability to communicate with other Controls on the network that are connected to Horizon by sending messages. Messages are sent from *Chat Windows* and are distributed to every Control listed in its corresponding *Chat Session*. The Chat module supports one-to-one communication with another Control, or group communication by inviting other Controls to the Chat Session from the *Invite Dialog*.

8.1 Chat Windows

The Chat Window sends and receives messages with Controls in its corresponding *Chat Session*. The Chat Window consists of the *Messages Panel* and the *Message Text Area*.

8.1.1 Chat Session

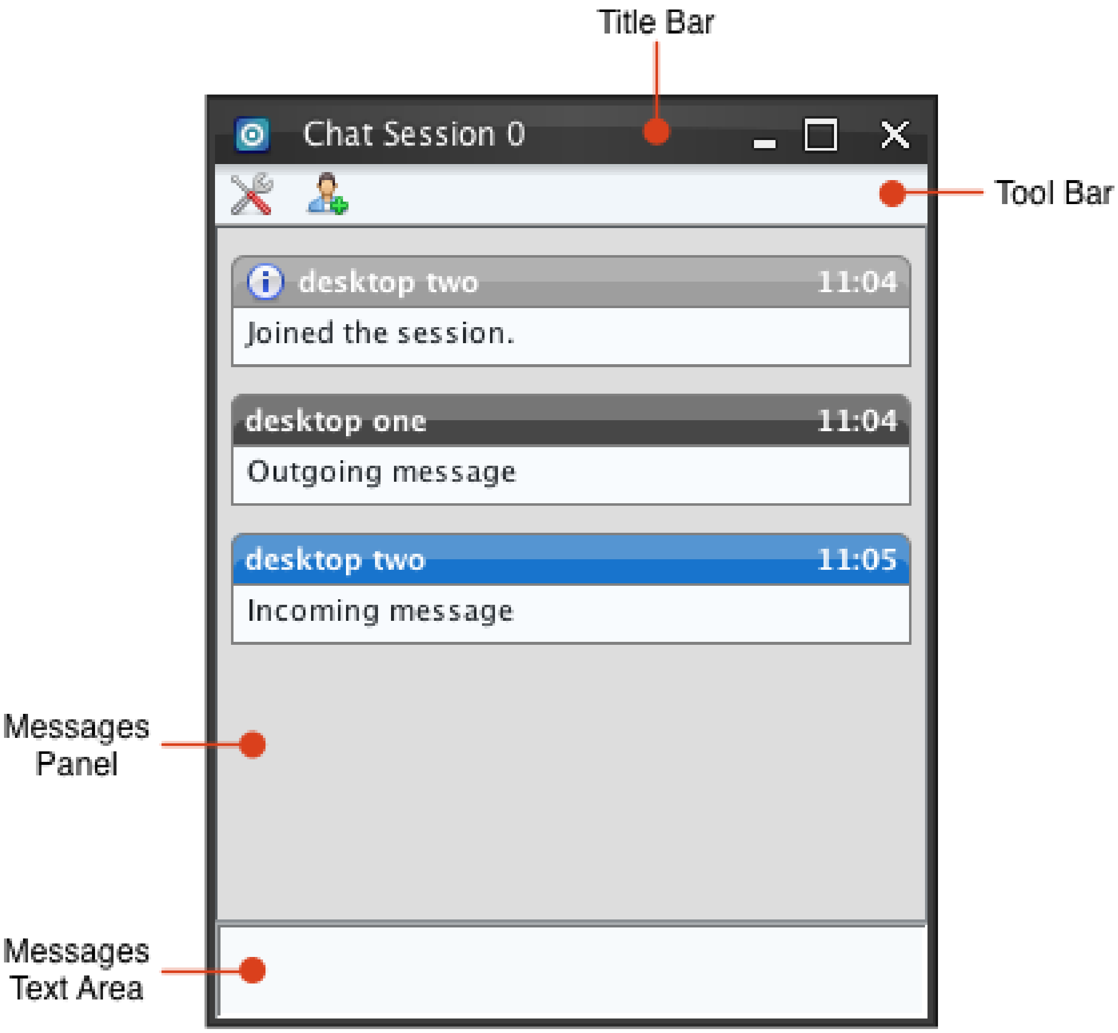
A Chat Window is created by initiating a Chat Session. A Chat Session is a list of Controls on the network participating in a conversation. Controls can be added or removed from a session. Chat Sessions are identified sequentially beginning with the number zero. The Chat Session's identification number is displayed in the Chat Window's Title Bar. A Chat Window represents only one Chat Session. Multiple Chat Sessions can be initiated, resulting in the creation of multiple Chat Windows.

Chat Sessions are initiated by right-clicking on the chosen Control and selecting "Session" from the Chat menu. This will launch a Chat Window on the chosen Control. If the chosen Control is connected to Horizon, it will automatically open a Chat Window and be joined to the session. If the chosen Control is not connected to Horizon, the local Control's Chat Window will remain in a disabled state.

8.1.2 Messages Panel

All messages are added to the Messages Panel. Three types of messages can be added to the Messages Panel:

- Incoming Messages: Message received from another Control. Incoming Messages will appear with a blue header.
- Outgoing Messages: Message sent from the local Chat Window. Outgoing Messages will appear with a dark gray header.
- Status Messages: Status messages are displayed when 1) a Control is added to the session, 2) a Control leaves the session, or 3) a Control is typing. Status Messages will appear with a light gray header and the ℹ icon.



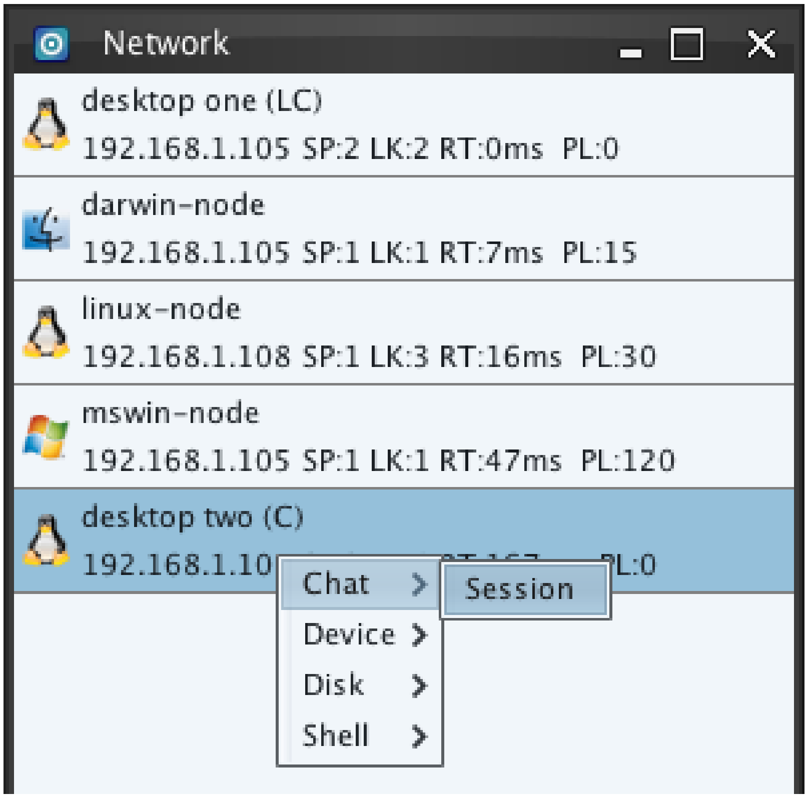
desktop two 11:04
Joined the session.

desktop one 11:04
Outgoing message

desktop two 11:05
Incoming message

Fig. 8.1: **Chat Window**

]



desktop one (LC)
192.168.1.105 SP:2 LK:2 RT:0ms PL:0

darwin-node
192.168.1.105 SP:1 LK:1 RT:7ms PL:15

linux-node
192.168.1.108 SP:1 LK:3 RT:16ms PL:30

mswin-node
192.168.1.105 SP:1 LK:1 RT:47ms PL:120

desktop two (C)
192.168.1.10 PL:0

Fig. 8.2: **Chat Popup Menu**

Every message includes the time of reception, in 24-hour format, on the right-hand side of the message header. Incoming and Outgoing Messages will display the name of the relevant Control on the left-hand side of the message header. Status messages will display the name of the relevant Control if the Control is joining or leaving the *Chat Session*.

8.1.3 Message Text Area

The Message Text Area sends a text message to every Control in the *Chat Session*. To send a message, type into the Message Text Area and press enter.

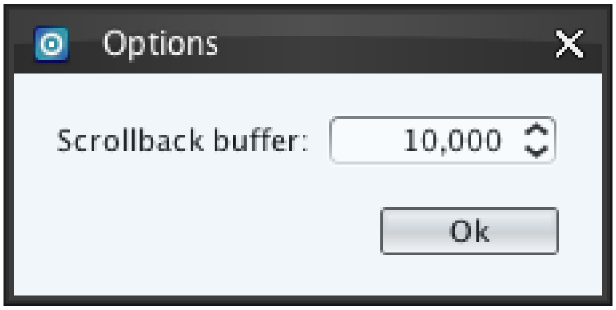
Note: A maximum 10,000 characters can be written to the Message Text Area. Attempting to paste in text with more than 10,000 characters will be denied.

As a message is being typed, a status message will appear on every other Control in the session that states which Control is typing. Any text beginning with the "#" character will not send a status message. Once a status message is sent, it will remain on every Control in the session until the message is sent by pressing enter.

8.1.4 Options Dialog

Opens the Options Dialog. The Options Dialog consists of one field:

- Scrollback Buffer: Represents the maximum number of characters that can appear in the *Messages Panel*. The minimum value is 5,000, and the maximum value is 50,000. If the number of characters exceeds the Scrollback Buffer value, the first message(s) will be truncated so that the number of characters equals the Scrollback Buffer value. Messages that are truncated will begin with ”...”. Messages are truncated before a new message appears on the *Messages Panel*, so that the most recent message will always display in its entirety.

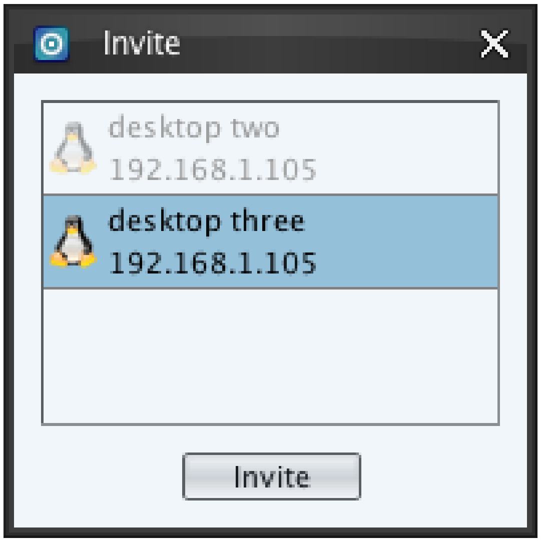


Scrollback buffer: 10,000

Fig. 8.3: **Options Dialog**

8.1.5 Invite Dialog

Opens the Invite Dialog. The Invite Dialog will list every Control on the network. Controls that are participating in the session will appear disabled (semi-transparent), and cannot be selected. To invite a Control to the session, select the Control from the list and click the Invite button. Multiple Controls can be selected simultaneously.



desktop two
192.168.1.105

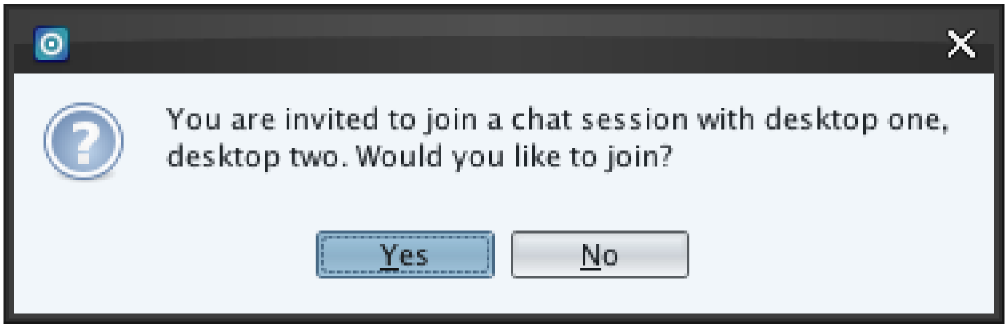
desktop three
192.168.1.105

Fig. 8.4: **Invite Dialog**

The Control being invited will display an Invite Prompt asking the user to join the *Chat Session*. The Invite Prompt will include the names of Controls that are listed in the session. To join the session, click Yes.

8.1.6 Session Participants

There are two ways to discover which Controls are participating in the session.



You are invited to join a chat session with desktop one, desktop two. Would you like to join?

Fig. 8.5: **Invite Prompt**

- The first way is to type “#users” in the Message Text Area and press enter. A status message will be added to the *Messages Panel* listing which Controls are in the session.
- The second way is to open the *Invite Dialog*. Any Control participating in the session will appear disabled.

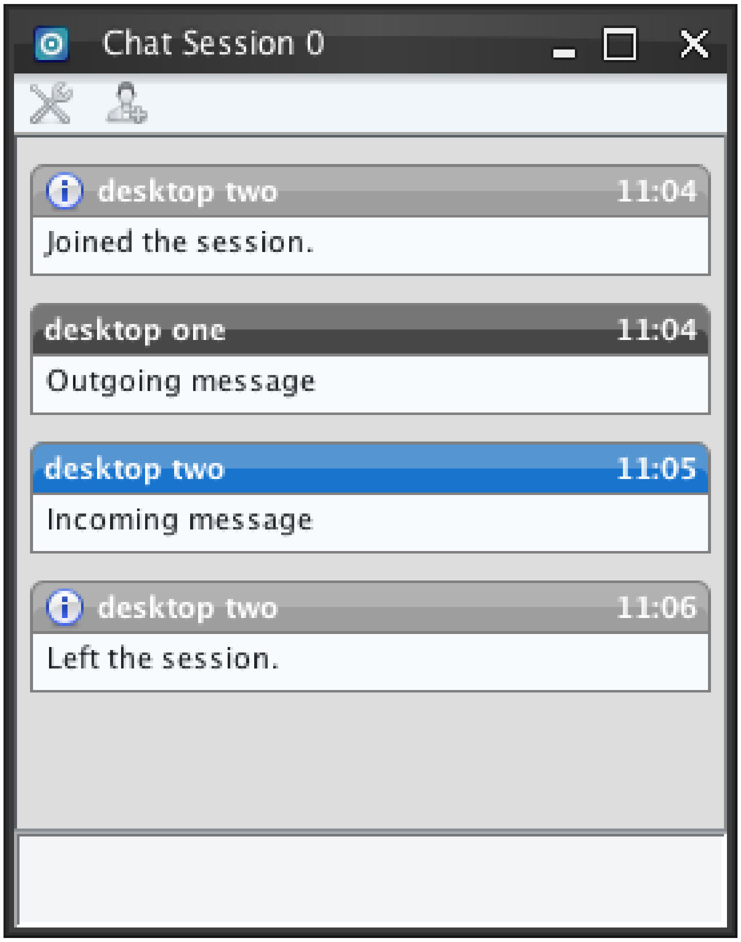
8.1.7 Exiting a Session

To leave a session, simply close the Chat Window. The remaining Controls in the session will display a status message stating which Control has left the session. If only one Control remains in the session, the session will be closed and the Chat Window will change to a disabled state. Once the session is closed, no Controls can be added to the session.

8.2 Chat Session Log Files

All messages are automatically logged in its base directory in the subfolder chat-session-logs. Every session is given its own log file with the date the session was initialized as the file name and ”.log” as the extension. Incoming Messages and Outgoing Messages are written in the format: “[time] [control name]> [message]”. Status messages are written in the format: “[time] [message]”. Typing Status Messages are not logged. Below is an example of a log file:

```
12:10:01 desktop two has joined the session.
12:10:15 desktop one>Hello, desktop two.
12:10:25 desktop two>Hello.
12:10:25 desktop two>Goodbye.
12:10:30 desktop two has left the session.
```



desktop two 11:04
Joined the session.

desktop one 11:04
Outgoing message

desktop two 11:05
Incoming message

desktop two 11:06
Left the session.

Fig. 8.6: **Closed Chat Session**

CHAPTER

NINE

LOADER MODULE

The Loader Module provides the ability to remotely load Microsoft Windows DLLs within the address space of a Node. In release v3.1.0 Loader Module has been upgraded to support ICE specification version 3. Four loading behaviors are supported by DLLs: Fire, Fire and Forget, Fire and Collect, Fire and Interact. For brevity these are referred to from here on as Fire, Forget, Collect, and Interact respectively.

9.1 Deployment Requests

DLLs are staged in a cache on the Control prior to deploying. Deploying a DLL requires staging it first then executing the following requests from Horizon or the API in the order listed. The Horizon automatically sends each request when a DLL is deployed using its Call Dialog.

1. Validate: Validates the DLL and a target Device against the DLL's metadata file.
2. Attach: Connect the DLL to a Unix Domain Socket on the Control for Collect or Interact behaviors. This request is skipped for Forget and Fire behaviors.
3. Load: Transfers and loads the DLL into memory on the target Device.
4. Start: Starts the DLL. This step sends the command line, executes DLLMain, and executes the MemoryLoad function per ICE specification.

9.2 Loader Window

The Loader Window is a global window and can be accessed from the Dashboard's Module combo box when the Local Control has the Loader Module activated. This window contains the *Staged Panel* and the *Loaded Panel*, which allows the operator to view a list of libraries.

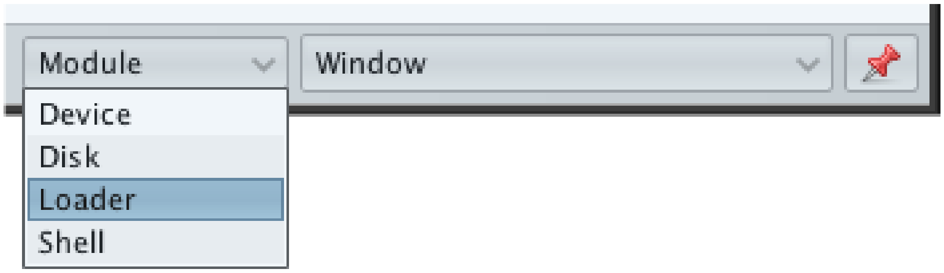
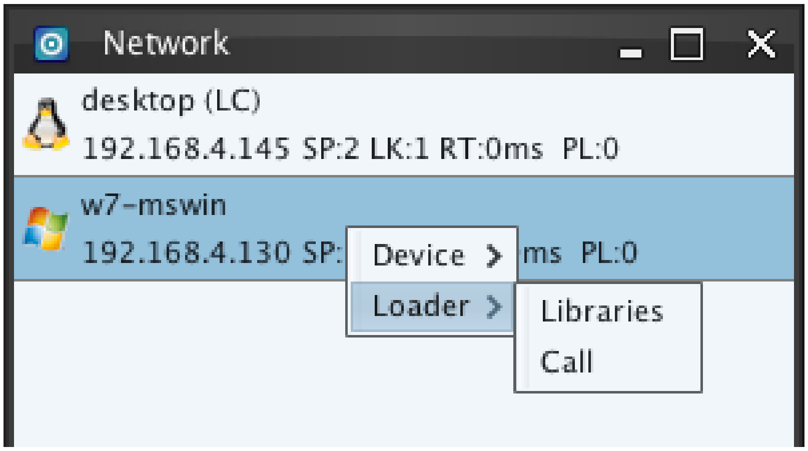


Fig. 9.1: **Dashboard Module Menu**

The Loader Window can also be opened by right clicking on a Node with the Loader Module activated and selecting "Libraries". The Loader Window includes Loaded and Staged Panels which are selected using the Side Panel.



desktop (LC)
192.168.4.145 SP:2 LK:1 RT:0ms PL:0

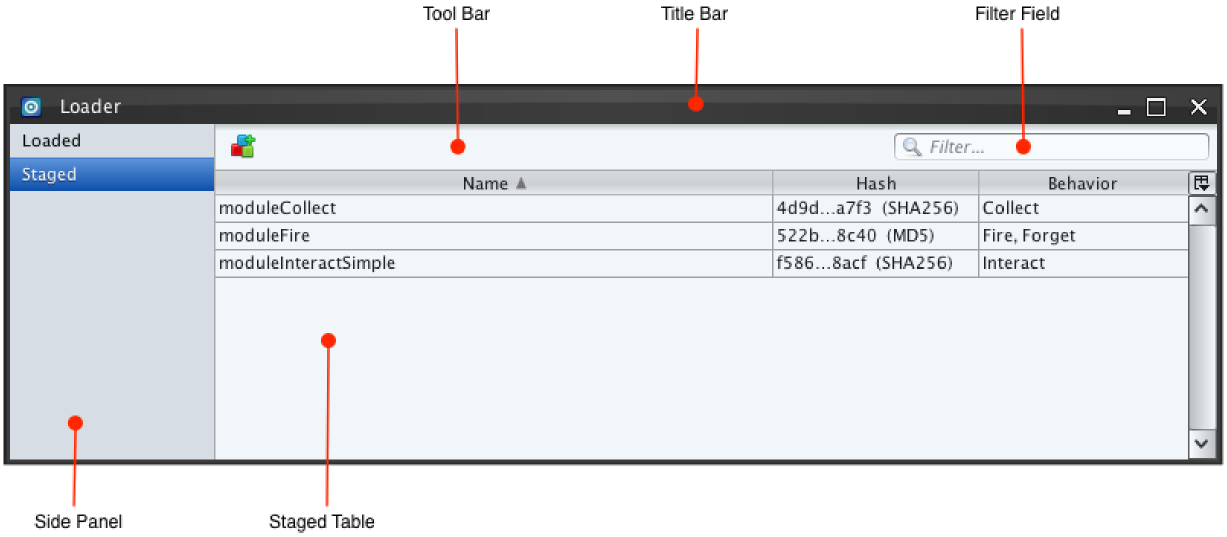
w7-mswin
192.168.4.130 SP: ms PL:0

Fig. 9.2: **Node Context Menu**

9.2.1 Staged Panel

The Staged Panel is automatically displayed when the Loader Window is open. The Staged Panel shows a list DLLs uploaded from Horizon or API clients to the Control. Press "Staged" in the Side Panel to display the Staged Panel.

To Stage a library press the Stage button . A file dialog appears to select a DLL to stage. DLLs must have their META.xml metadata files present in the same directory as the DLL. The metadata file must be named the same as the DLL with the extension "META.xml" appended. These metadata requirements follow the version 3 specification.



Name ▲	Hash	Behavior
moduleCollect	4d9d...a7f3 (SHA256)	Collect
moduleFire	522b...8c40 (MD5)	Fire, Forget
moduleInteractSimple	f586...8acf (SHA256)	Interact

Fig. 9.3: **Staged Panel**

The columns of the Staged Table are as follows:

- Name: The DLL filename with the file extension removed.
- Hash: The MD5 or SHA256 hash from the metadata file printed in an abbreviated format showing the first and last four hexadecimal characters along with (MD5) or (SHA256).
- Behavior: The behaviors the DLL can be deployed as. This is taken from the "featureset" entry of the DLL's metadata file.

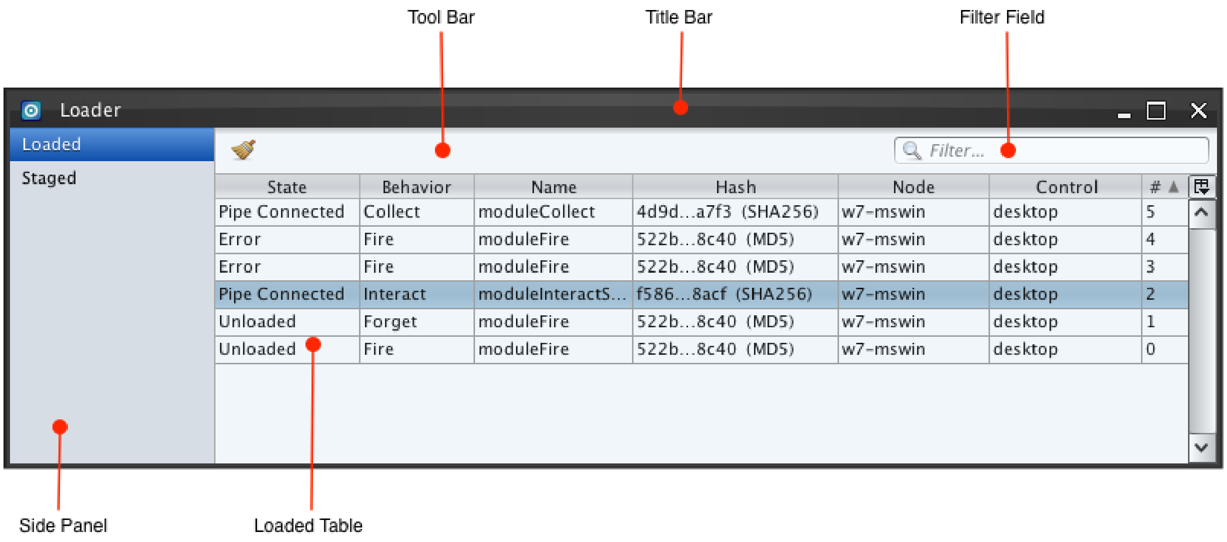
If staging is successful a new entry will appear in the Staged Panel. The Control stores cached DLLs and metadata files inside its base directory with the DLL file's MD5 included in the filename:

```
basedir.0
|-- device.db
|-- device.log
|-- loader-cache
|    |-- moduleCollect-30fa4c2af50b371d811e55500e731539.dll
|    |-- moduleCollect-30fa4c2af50b371d811e55500e731539.META.xml
|    |-- moduleFire-522b9d352b962a2aec2eae169e2f8c40.dll
|    |-- moduleFire-522b9d352b962a2aec2eae169e2f8c40.META.xml
|    |-- moduleInteractSimple-c0d24bf18d3a0f325d7345843b0148f0.dll
|    |-- moduleInteractSimple-c0d24bf18d3a0f325d7345843b0148f0.META.xml
|-- loader.log
```

To remove a DLL from the staging cache, right click on the item in the Staged Table and select "Unstage" from the popup menu.

9.2.2 Loaded Panel

The Loaded Panel is selected be clicking "Loaded" in the Side Panel in the Loader Window. Deployed DLL instances populate the Loaded Table.



State	Behavior	Name	Hash	Node	Control	# ▲
Pipe Connected	Collect	moduleCollect	4d9d...a7f3 (SHA256)	w7-mswin	desktop	5
Error	Fire	moduleFire	522b...8c40 (MD5)	w7-mswin	desktop	4
Error	Fire	moduleFire	522b...8c40 (MD5)	w7-mswin	desktop	3
Pipe Connected	Interact	moduleInteractS...	f586...8acf (SHA256)	w7-mswin	desktop	2
Unloaded	Forget	moduleFire	522b...8c40 (MD5)	w7-mswin	desktop	1
Unloaded	Fire	moduleFire	522b...8c40 (MD5)	w7-mswin	desktop	0

Fig. 9.4: **Loaded Panel**

The Loaded Table contains the following columns:

- State: The deployed DLL instance's runtime state. The possible states are listed below.

State	Meaning
Validated	The validation request succeeded. A pre-deployment DLL instance exists on the Local Control.
Attaching	Validation request succeeded and a request to attach to the Control's Unix Domain Socket is pending. The DLL instance is still in pre-deployment.
Attached	Attach request has succeeded. The Unix Domain Socket is connected on the Control. The DLL instance is still in pre-deployment.
Transferring	Load request hash been issued and the DLL is being sent to the target Device.
Transferred	The DLL has arrived at the target Device.
Loading	The DLL has begun loading. The target Device starts loading automatically after transferring.
Loaded	The DLL has been loaded into memory successfully.
Activating	A start request has been sent and has arrived at the target Device and DLLMain is being called.
Activated	DLLMain has returned.
Starting	The exported MemoryLoad function has been called.
Completed	MemoryLoad function has completed.
C2 Down	This state is reported when the target node is offline from the Control. The state will change back to Pipe Connected if the node comes back online before the disconnect timeout is reached. This state only applies to Collect and Interact behaviors. See the *disconnect timeout* in the Call Dialog.
Resuming	This state is reported when an offline target node comes back online while communications are resynchronized before resuming back to Pipe Connected.
Pipe Connected	The named pipe on the target Device has connected. This state only comes after the Completed State. This state is only for Collect or Interact behaviors.
Unloading	The DLL has begun unloading on the target Device, either by request or on its own.
Unloaded	The DLL has unloaded from the target Device.
Nonresponsive	The DLL had the willQuit flag set in its metadata file. The DLL instance was signaled to unload. The willQuit flag causes the unloading request to complete without waiting on the DLL to fully unload. This is a special final state to indicate that Fluxwire has quit tracking the DLL instance. The DLL instance may or may not be running still.
Halted	The halted state is sent if the Fluxwire Loader plugin is unloaded from the target Device while a DLL is loaded or running. This state is also used during target Device shutdown.
Node Down	The target Device is offline from the Control anymore. The state of the DLL instance is unknown to the Control.
Exited while Offline	The target Device went offline but then came back online. However, either the DLL instance exited on its own, or the disconnect timeout was reached and the device forced the DLL to unload.
Error	There has been an error. The error message is in the Logs Window and in the Call Dialog if it is still open from deploying the DLL.

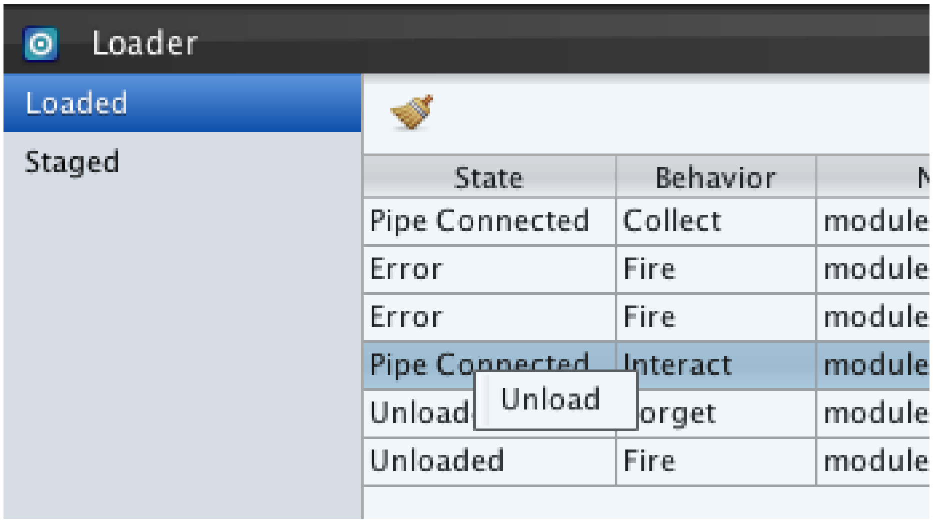
- Behavior: The behavior type deployed on each loaded DLL instance: Fire, Forget, Collect, or Interact.

- Name: The original filename of the DLL without the extension.
- Hash: The hashcode specified in the staged DLL's metadata file.
- Node: The name of the target Device.
- Control: The name of the Control that deployed the DLL instance. This Control supplies the Unix Domain Socket for Collect or Interact behaviors.
- #: A unique instance number for the deployed DLL. This number is unique only to the Control that deployed the DLL.

Clears all successfully completed DLL instances from the Loaded Table. This may leave behind records that are in a finalized state such as (Error, Node Down, Unloaded, Halted, or Nonresponsive).

Collect or Interact instances will run in the "Pipe Completed" state once successfully started and C2 connected. Forget or Fire instances run in the "Completed" state until they begin unloading.

Loaded DLL instances can be unloaded by right clicking the instance in the Loaded Table and selecting "Unload". Horizon allows operators to issue Unload requests to any instance regardless of state. If the instance cannot be unloaded, the Control or target Device responds with an appropriate error message. In general Collect or Interact behaviors can be signaled to unload in the Loaded, Activating, Activated, Starting, Completed, or Pipe Connected states. The unload request may complete before the DLL is fully unloaded if the willQuit flag was set in the metadata file, as according to the ICE specification.



State	Behavior	M
Pipe Connected	Collect	module
Error	Fire	module
Error	Fire	module
Pipe Connected	Interact	module
Unload	orget	module
Unloaded	Fire	module

Fig. 9.5: **Unload menu**

9.2.3 Call Dialog

A staged DLL is deployed from the Call Dialog in Horizon. The Call Dialog can be opened in two ways. The first is to right click a staged DLL in the Staged Table and select "Call" from the popup menu. The second way is to select the target Device in the Network Window and select "Call" from the Loader submenu.

The Call Dialog auto-selects the staged DLL or the target Device depending on where the menu "Call" was selected from. The Device and Module combo boxes automatically filter incompatible entries. For instance, if a DLL is selected

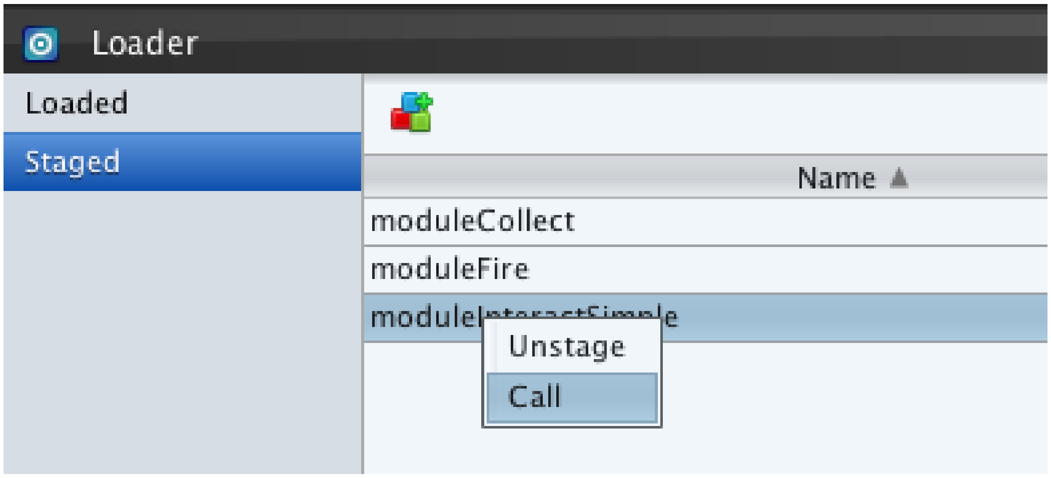


Fig. 9.6: **Right click Call menu item from Staged Table**

in the Module combo box that does not support a Node running on Windows XP, then that Node will not appear in the Device combo box until a different Module is selected.

Once the DLL is selected in the Module Combo box the Behavior ComboBox and Arguments Textfield will auto populate with data from the DLL's metadata file. The Arguments field's auto-complete populates with the exported function command line prefixes from the DLL's metadata file. Finally, the bottom status area will show any exported functions from the metadata file and display a notice if the willQuit flag is not set.

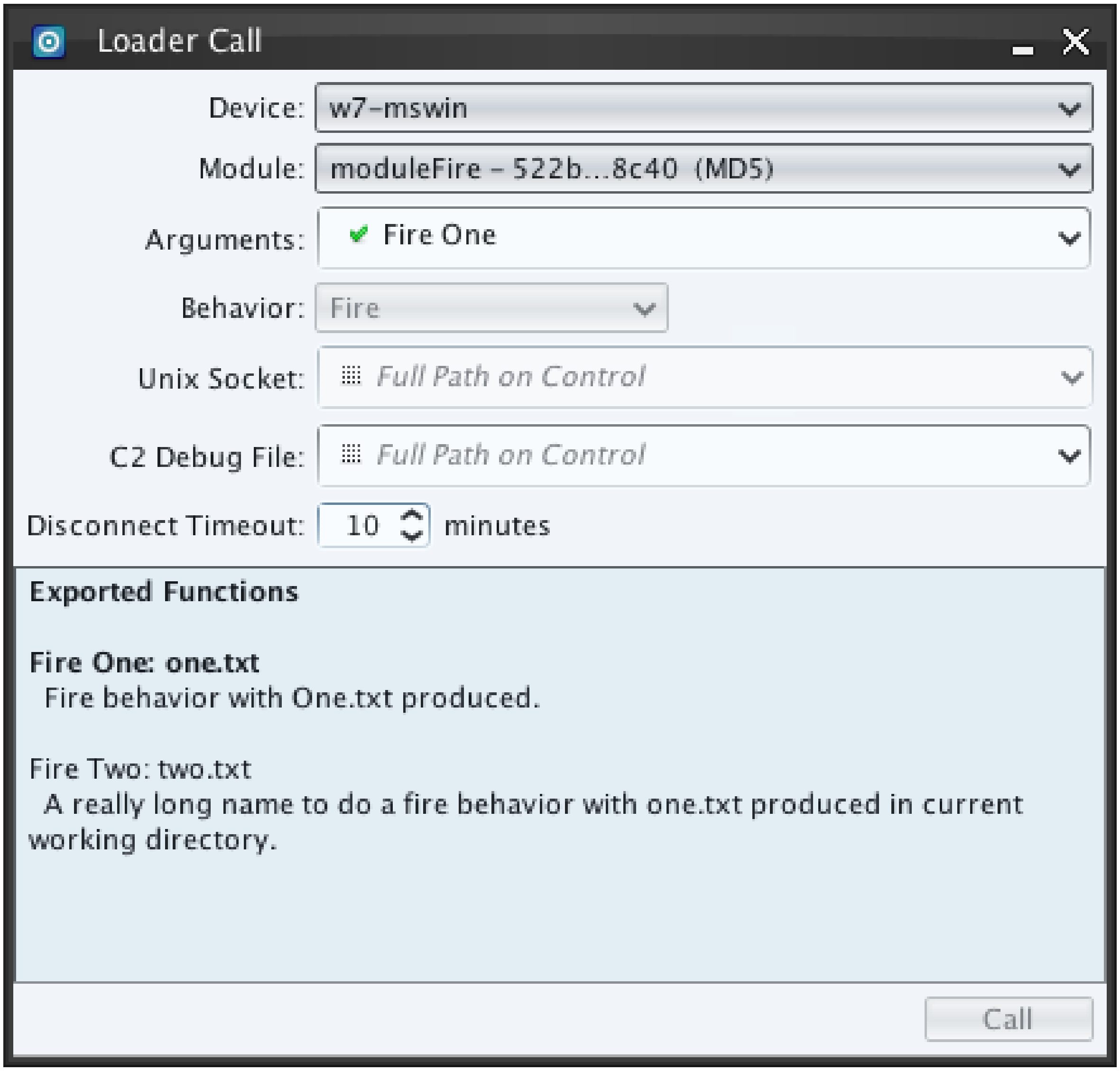
The Unix Socket Textfield is used to specify a full path to the C2 on the Control for Collect or Interact behaviors. This field is not used for Forget or Fire behaviors.

The C2 Debug File is optional for Collect or Interact behaviors. It accepts a full path to be specified on the Control to save a pcap format file to dump all the data read or written to the Unix Socket on the Control. The generated file can be later loaded into tools like Wireshark to debug data passing end to end on the ICE C2 for that DLL instance. The Disconnect Timeout is an optional field for Collect or Interact behaviors. It specifies how long in minutes the loaded DLL instance should wait before tearing down if the Fluxwire device go offline from its Control. Instances loaded on offline devices will show up in the Loaded Panel with "C2 Down" state. If the disconnect timer expires before the node comes back online the final state value will be "Exited while Offline". When a offline device comes back online a "Resuming" state is reported to API and Horiozn while C2 communications resynchronizes.

The Call button is enabled when valid values are selected in all required fields. Once pressed, the status area is cleared and the deployment steps are automatically added to the status area as each request succeeds. If a failure or error is returned, the deployment stops.

Note: The Call Dialog does not need to remain open after deployment starts. Horizon forbids it to be closed before the first response to the Validate request is received. After validation, the Call Dialog can be closed and Horizon will continue to automatically deploy the DLL instance.

Warning: The Loader Module must be autoloaded on Controls in order for multiple clients (Horizon or Control API) instances to work correctly. The *Desktop* configuration file autoload local stanza should include "loader".



Device: w7-mswin

Module: moduleFire – 522b…8c40 (MD5)

Arguments: ✓ Fire One

Behavior: Fire

Unix Socket: Full Path on Control

C2 Debug File: Full Path on Control

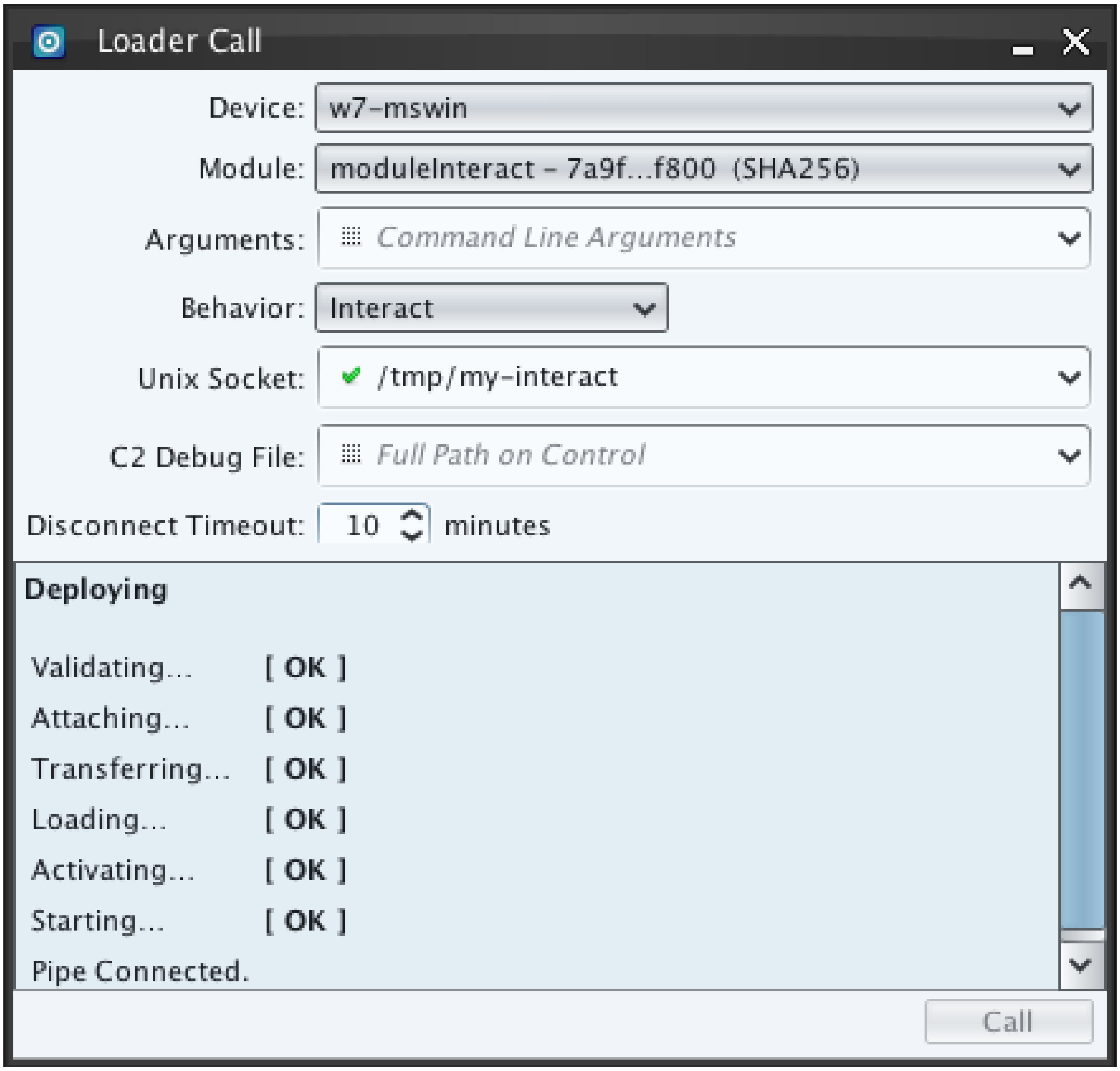
Disconnect Timeout: 10 minutes

Exported Functions

Fire One: one.txt
Fire behavior with One.txt produced.

Fire Two: two.txt
A really long name to do a fire behavior with one.txt produced in current working directory.

Fig. 9.7: **Call Dialog**



Loader Call

Device: w7-mswin

Module: moduleInteract – 7a9f...f800 (SHA256)

Arguments: Command Line Arguments

Behavior: Interact

Unix Socket: ✓ /tmp/my-interact

C2 Debug File: Full Path on Control

Disconnect Timeout: 10 minutes

Deploying

```
Validating...     [ OK ]
Attaching...      [ OK ]
Transferring...   [ OK ]
Loading...        [ OK ]
Activating...     [ OK ]
Starting...       [ OK ]
Pipe Connected.
```

Fig. 9.8: **Call Dialog status updates while deploying**

CHAPTER

TEN

TRANSPORT MODULE

The transport module implements a sophisticated Virtual Private Network (VPN) capability for Fluxwire mesh networks. The module supports application layer packets for TCP and UDP protocols.

10.1 Virtual Interface

The VPN is exposed to the operator's workstation through a virtual interface. During startup the virtual interface is allocated and configured by the settings specified in the Fluxwire configuration file.

10.2 Transport Window

The Transport Window is a global window accessible from the Dashboard's Module combo box when the Local Control has the Transport module activated.

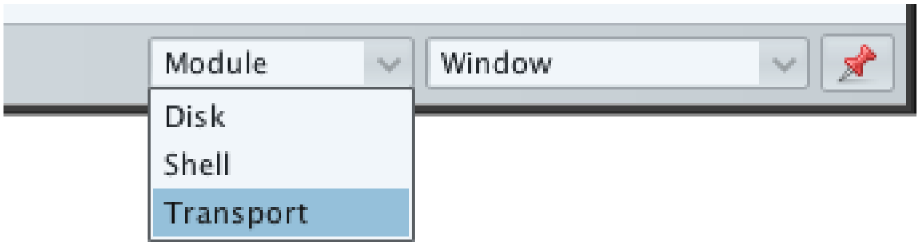


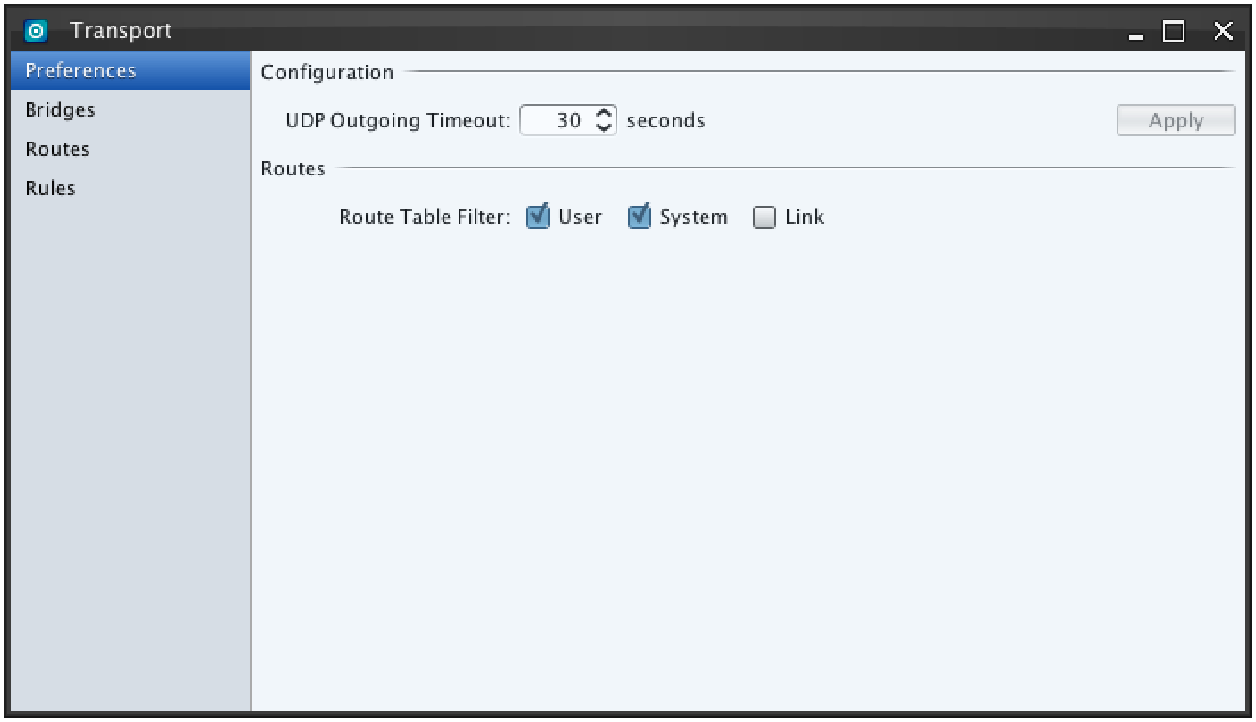
Fig. 10.1: **Module Combo Box**

10.2.1 Preferences

The Preferences panel provides the ability to adjust the UDP outgoing timeout and filter the types of *Routes* displayed.

UDP Outgoing Timeout

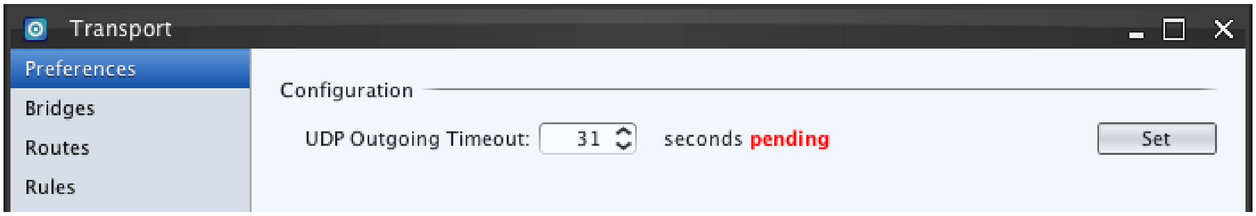
If the UDP outgoing timeout is changed a 'pending' indication appears until the "Set" button is clicked to confirm the new timeout.



UDP Outgoing Timeout: 30 seconds

Route Table Filter: ☑ User ☑ System □ Link

Fig. 10.2: **Preferences Panel**



UDP Outgoing Timeout: 31 seconds pending

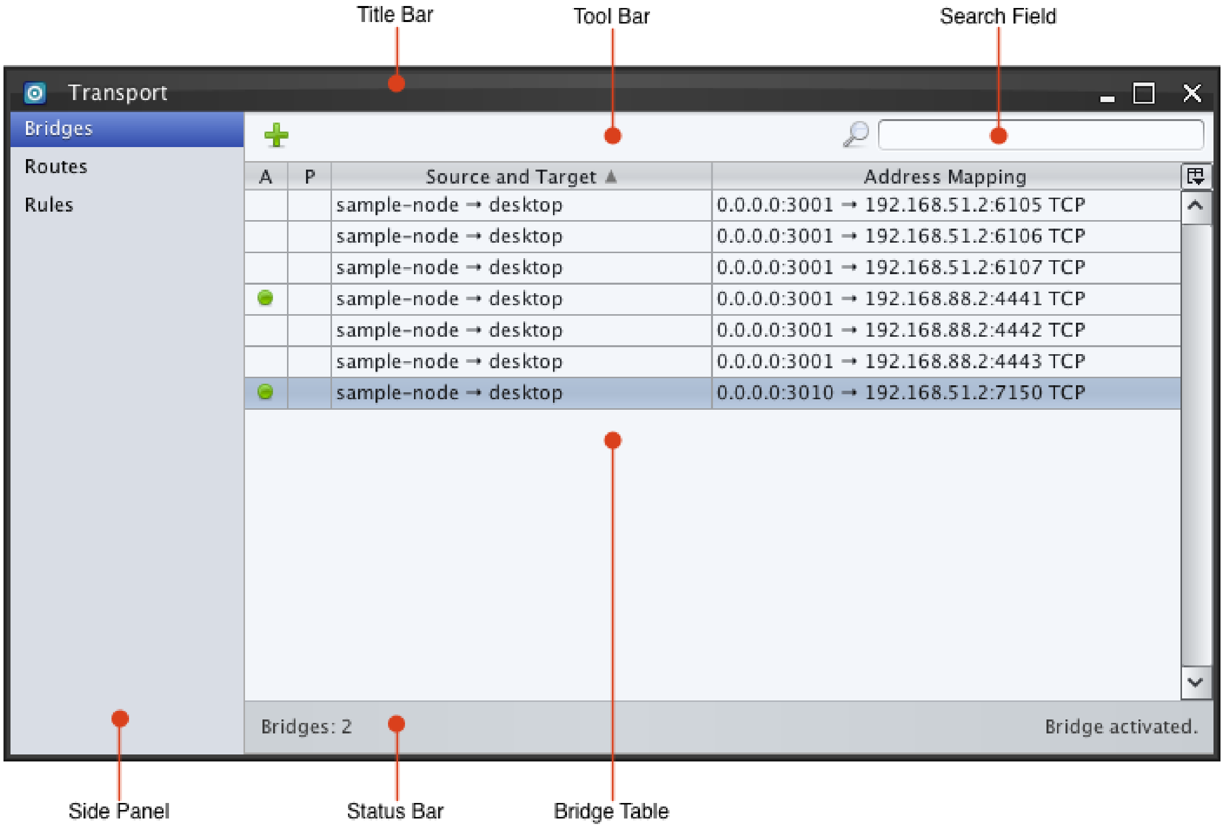
Fig. 10.3: **UDP Outgoing Timeout**

10.2.2 Bridges

Bridges provide an advanced port forwarding capability. This is similar to the one in SSH (Secure Shell) but using the distributed features of the VPN. Bridges expand traditional port forwarding capability with the following additional features:

- Multi-hop connections across the mesh network.
- Port forwarding between two distinct Nodes or a Node and Control.
- Convenient setup and configuration.

Traffic arrives at the source port on a specified source device and is funneled to the target device leaving from the specified target device's port. The Bridge panel displays when the operator selects the Bridges item in the side panel of the Transport Window.



A	P	Source and Target ▲	Address Mapping
		sample-node → desktop	0.0.0.0:3001 → 192.168.51.2:6105 TCP
		sample-node → desktop	0.0.0.0:3001 → 192.168.51.2:6106 TCP
		sample-node → desktop	0.0.0.0:3001 → 192.168.51.2:6107 TCP
●		sample-node → desktop	0.0.0.0:3001 → 192.168.88.2:4441 TCP
		sample-node → desktop	0.0.0.0:3001 → 192.168.88.2:4442 TCP
		sample-node → desktop	0.0.0.0:3001 → 192.168.88.2:4443 TCP
●		sample-node → desktop	0.0.0.0:3010 → 192.168.51.2:7150 TCP

Bridges: 2

Bridge activated.

Fig. 10.4: **Bridge Control Panel**

Bridge Table

The Bridge table is organized into four columns that describe the following:

- A (Active): Indicates if the bridge is active or inactive using a green ● or gray ● indicator respectively.
- P (Persistence): Indicates whether the bridge is in persistence mode. In this mode the bridge will continue to stay active long after the Local Control has been shutdown or disconnected.
- Source and Target: Identifies the source and destination devices by name.

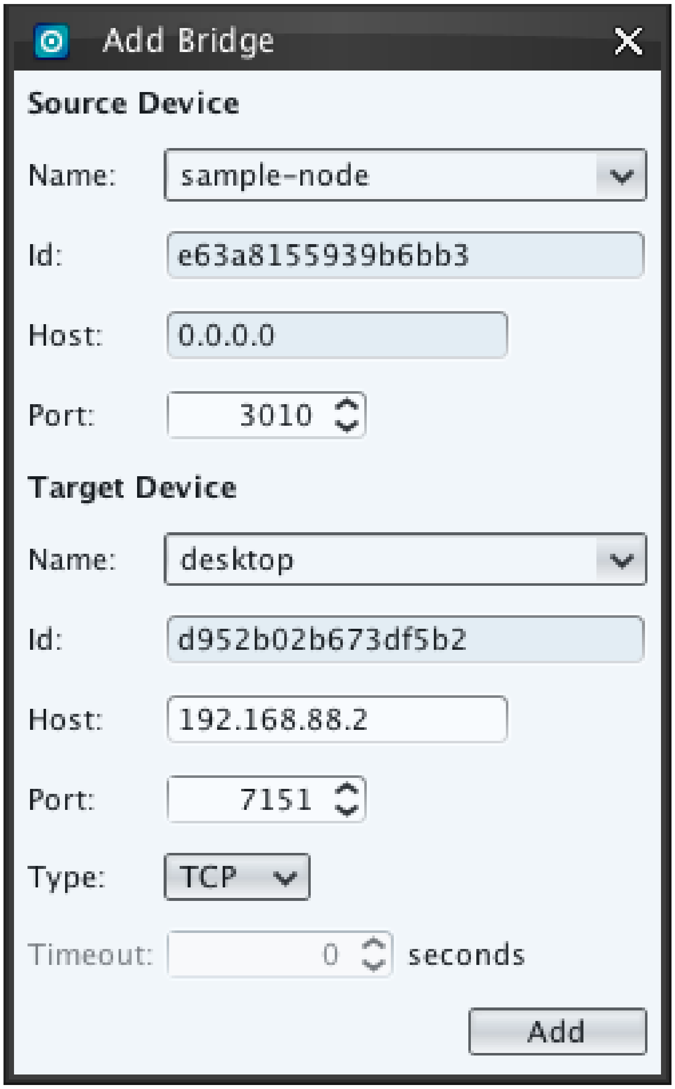
- Address Mapping: Identifies the address mapping between the source and the target device. From left to right, the source IP address, source port, remote IP address, remote port, and type.

The Bridge table exposes three menu options using the right click menu.

- Enable: Activates a bridge.
- Disable: Deactivates a bridge.
- Delete: Removes a disabled bridge's configuration from the table.

Bridge Configuration

➕ Using the "Add Bridge" button, the operator can add an inactive bridge.



Add Bridge

Source Device

Name: sample-node

Id: e63a8155939b6bb3

Host: 0.0.0.0

Port: 3010

Target Device

Name: desktop

Id: d952b02b673df5b2

Host: 192.168.88.2

Port: 7151

Type: TCP

Timeout: 0 seconds

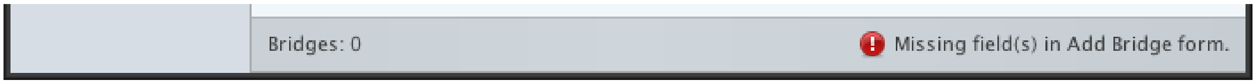
Fig. 10.5: **Add Bridge**

The "Add Bridge" dialog consists of the following components:

- Source Name: Name of the source device that will listen for incoming connections.

- Source Id: Hexadecimal id of the source device. (Read Only)
- Source Host: Always set to “0.0.0.0” to listen to all interfaces. (Read Only)
- Source Port: The listening port on the source device.
- Target name: Name of the target device that will initiate outbound connections on behalf of the source device.
- Target Id: Hexadecimal id of the target device. (Read Only)
- Target Host: The remote IP address for the outbound connection. Defaults to the virtual interface address of the Local Control.
- Target Port: The remote port for the outbound connection.
- Type: TCP or UDP connections.
- Target Timeout: Timeout associated with UDP connections, default value is 30 seconds.

If the Source Port, Target Host, or the Target Port are empty when the “Add” button is clicked an error will be displayed in the Bridge panel’s status bar:

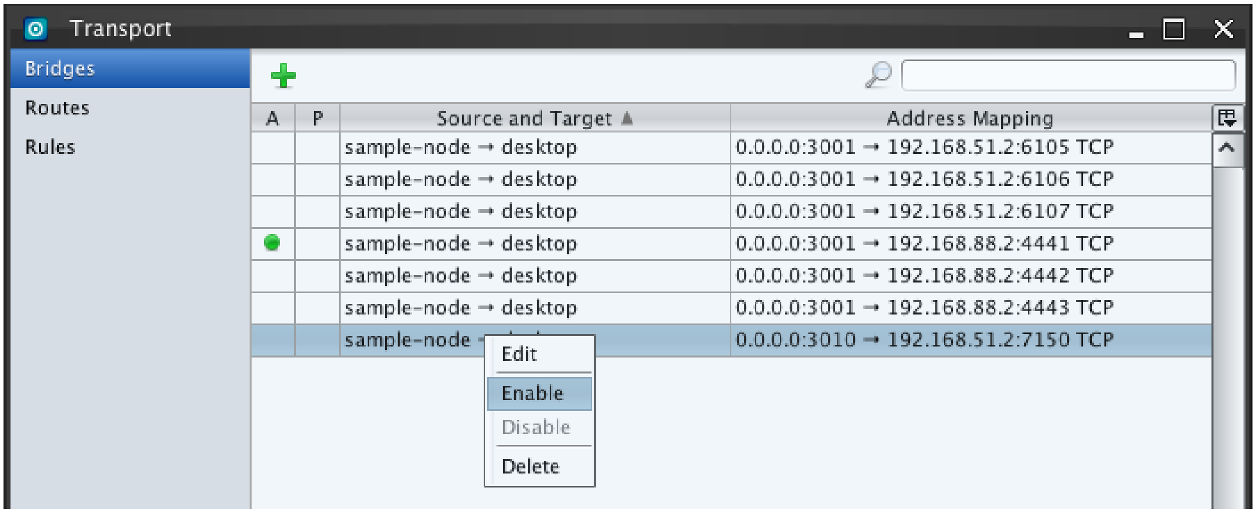


Bridges: 0

Missing field(s) in Add Bridge form.

Fig. 10.6: **Add Bridge Error**

Once the bridge is successfully added to the Bridge table, the operator can activate the bridge through the right click menu using the Enable option.

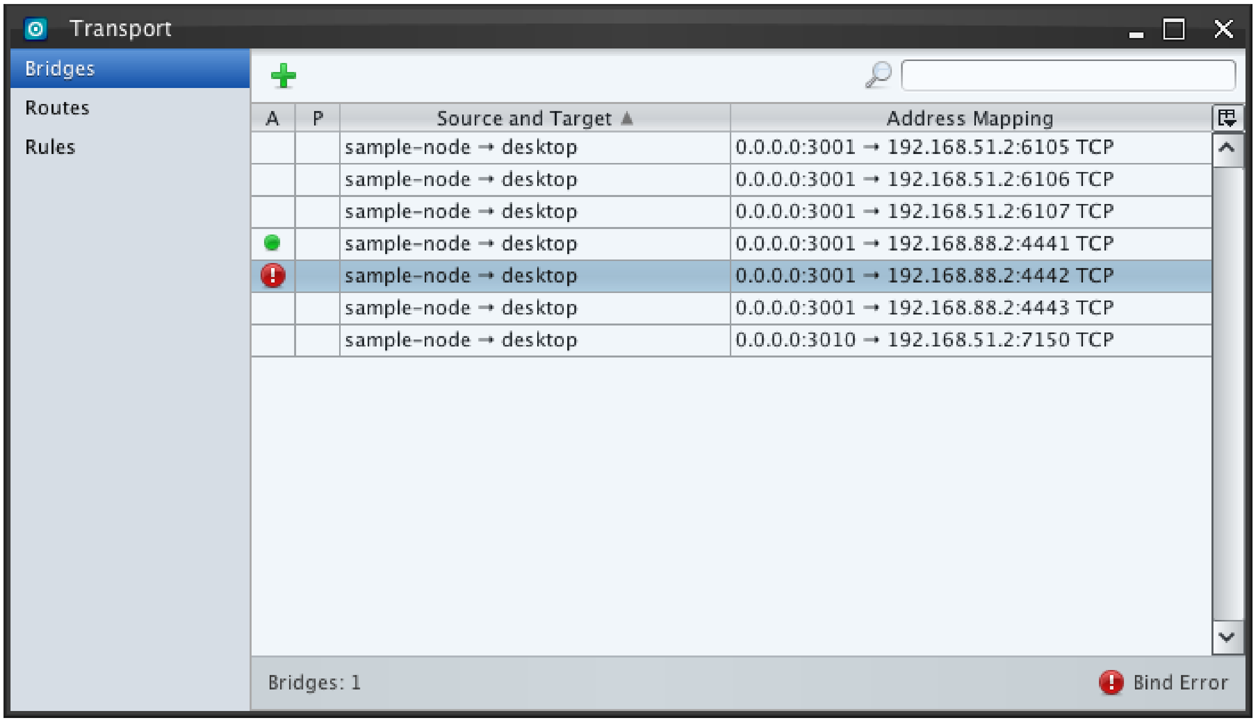


A	P	Source and Target ▲	Address Mapping
		sample-node → desktop	0.0.0.0:3001 → 192.168.51.2:6105 TCP
		sample-node → desktop	0.0.0.0:3001 → 192.168.51.2:6106 TCP
		sample-node → desktop	0.0.0.0:3001 → 192.168.51.2:6107 TCP
●		sample-node → desktop	0.0.0.0:3001 → 192.168.88.2:4441 TCP
		sample-node → desktop	0.0.0.0:3001 → 192.168.88.2:4442 TCP
		sample-node → desktop	0.0.0.0:3001 → 192.168.88.2:4443 TCP
		sample-node	0.0.0.0:3010 → 192.168.51.2:7150 TCP

Fig. 10.7: **Bridge Menu - Enable/Disable/Delete**

If a bridge cannot be enabled an error is displayed in the Bridge panel’s status bar. The screenshot below shows a bind error caused by attempting to bind to a port that has already been bound to by an active bridge.

Bridges must be inactive before their configurations can be modified. Once a bridge is disabled its configuration can be edited by right clicking and selecting the “Edit” option. The Edit Bridge dialog is identical to the *Add Bridge* dialog. Instead of creating a new row in the bridge table, the configuration of the bridge selected is modified.



A	P	Source and Target ▲	Address Mapping
		sample-node → desktop	0.0.0.0:3001 → 192.168.51.2:6105 TCP
		sample-node → desktop	0.0.0.0:3001 → 192.168.51.2:6106 TCP
		sample-node → desktop	0.0.0.0:3001 → 192.168.51.2:6107 TCP
●		sample-node → desktop	0.0.0.0:3001 → 192.168.88.2:4441 TCP
❗		sample-node → desktop	0.0.0.0:3001 → 192.168.88.2:4442 TCP
		sample-node → desktop	0.0.0.0:3001 → 192.168.88.2:4443 TCP
		sample-node → desktop	0.0.0.0:3010 → 192.168.51.2:7150 TCP

Bridges: 1

❗ Bind Error

Fig. 10.8: **Add Bridge Bind Error**

10.2.3 Routes

Routes enable access into the VPN by redirecting packets to the virtual interface. Route entries define the destination network addresses permitted to enter the VPN. Packets entering the virtual interface are processed and relayed to an operator configurable destination device, also referred to as the gateway device. Operators can select between full routing, partial routing, no routing. Full routing redirects all traffic into the VPN while partial routing selectively redirects some traffic. *Rules* evaluation selects the gateway device.

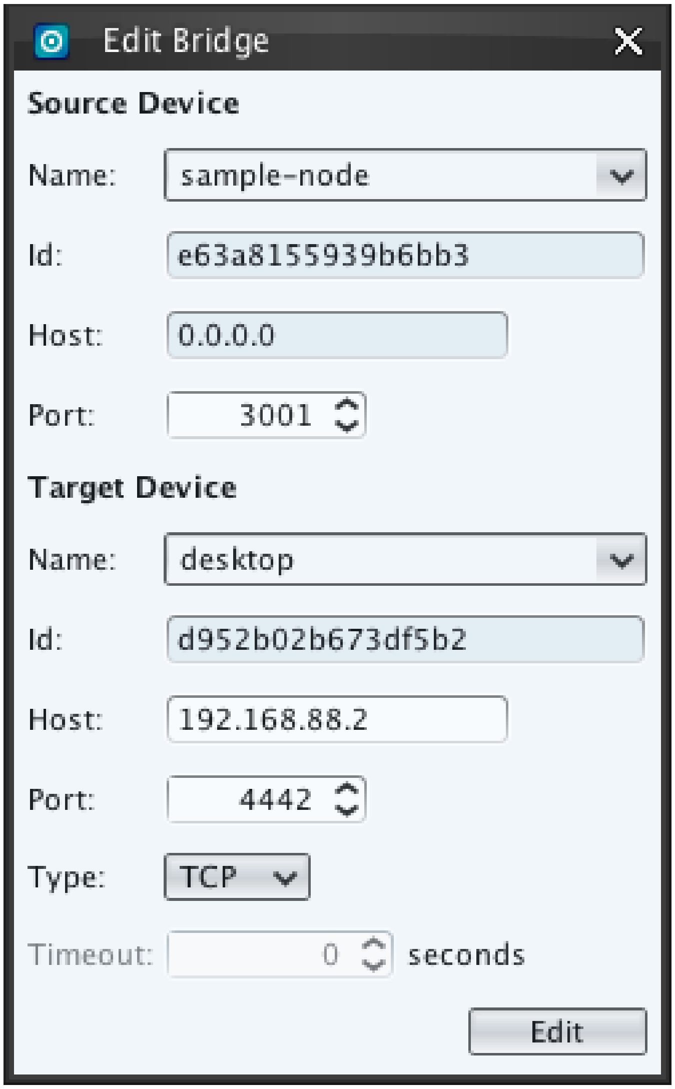
The Routes panel displays when the operator selects the Routes item in the side panel of the Transport Window.

Route Types

The transport module does not evaluate the routes, rather it manages them through the preexisting route tables in the operating system. There are three types of routes: User, Link, and System. User routes are routes created by operators using Horizon. User routes are created in a disabled state. Operators enable user routes using the Route Panel and the Local Control writes the entry in to the Operating System's route table. Likewise, when operators disable User routes the Local Control removes the corresponding entry from the Operating System's route table.

Link routes are automatically created and removed from the operating system's route tables by Local Controls when Nodes connect and disconnects to the Fluxwire network. Link routes are necessary so that Fluxwire's device links can propagate packets across the real network interface.

System routes are routes unrelated to Fluxwire that exist in the operating system's routing table. These routes are added by the operating system, other software applications, or manually by system administrators. The Route Panel allows operators to audit system routes of the operating system and will flag routes that may conflict with User and Link routes.



Edit Bridge

Source Device

Name: sample-node

Id: e63a8155939b6bb3

Host: 0.0.0.0

Port: 3001

Target Device

Name: desktop

Id: d952b02b673df5b2

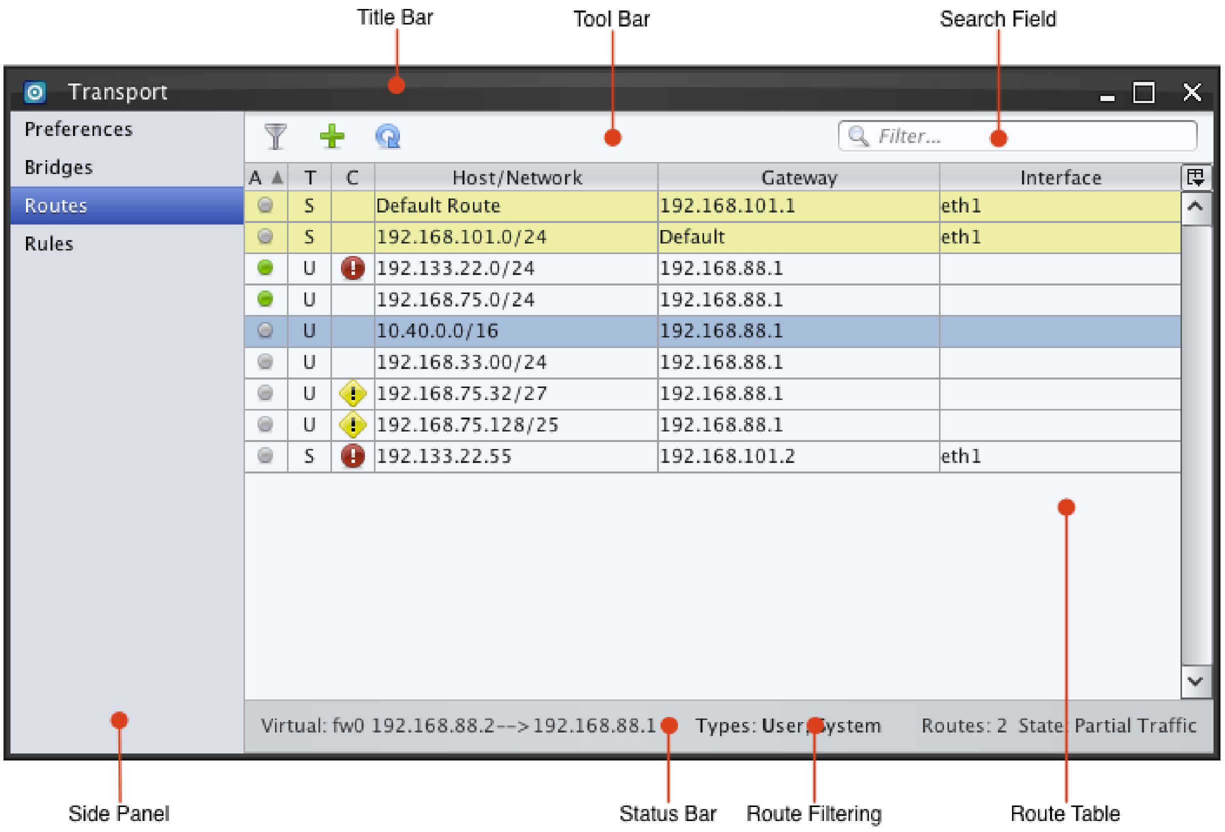
Host: 192.168.88.2

Port: 4442

Type: TCP

Timeout: 0 seconds

Fig. 10.9: **Edit Bridge**



A ▲	T	C	Host/Network	Gateway	Interface
○	S		Default Route	192.168.101.1	eth1
○	S		192.168.101.0/24	Default	eth1
●	U	❗	192.133.22.0/24	192.168.88.1	
●	U		192.168.75.0/24	192.168.88.1	
○	U		10.40.0.0/16	192.168.88.1	
○	U		192.168.33.00/24	192.168.88.1	
○	U	⚠	192.168.75.32/27	192.168.88.1	
○	U	⚠	192.168.75.128/25	192.168.88.1	
○	S	❗	192.133.22.55	192.168.101.2	eth1

Virtual: fw0 192.168.88.2-->192.168.88.1   Types: User, System   Routes: 2 State: Partial Traffic

Fig. 10.10: **Route Panel**

Virtual Interface and Gateway

The status bar displays the Transport module's virtual interface configuration as it was specified in the Local Control's (flx-desktop) configuration file. A connection that matches an active route with Gateway set to the virtual gateway's IP address are evaluated by the Transport Module's *Rules* to find which target Node for the connection leave the Fluxwire network from.

Routes Table

The Routes table displays all active and inactive route entries at runtime and is organized into six columns that describe the following:

- A (Active): A green 🟢 or gray ⚪ inactive status. An active status indicates the route is active and is being managed by Fluxwire. All Link routes display active statuses. User routes display active statuses once the operator enables them. System routes always display inactive statuses.
- T (Type): Displays U for User routes, L for Link routes, or S for System routes.
- C (Conflict): Indicates if this route is currently in conflict or would conflict with other active routes or system routes.

Symbol	Meaning
Blank	No conflict with any active routes.
⚠	The inactive route **would** conflict or has a configuration issue.
❗	The route is active and **does** conflict with other active routes or has a configuration issue.

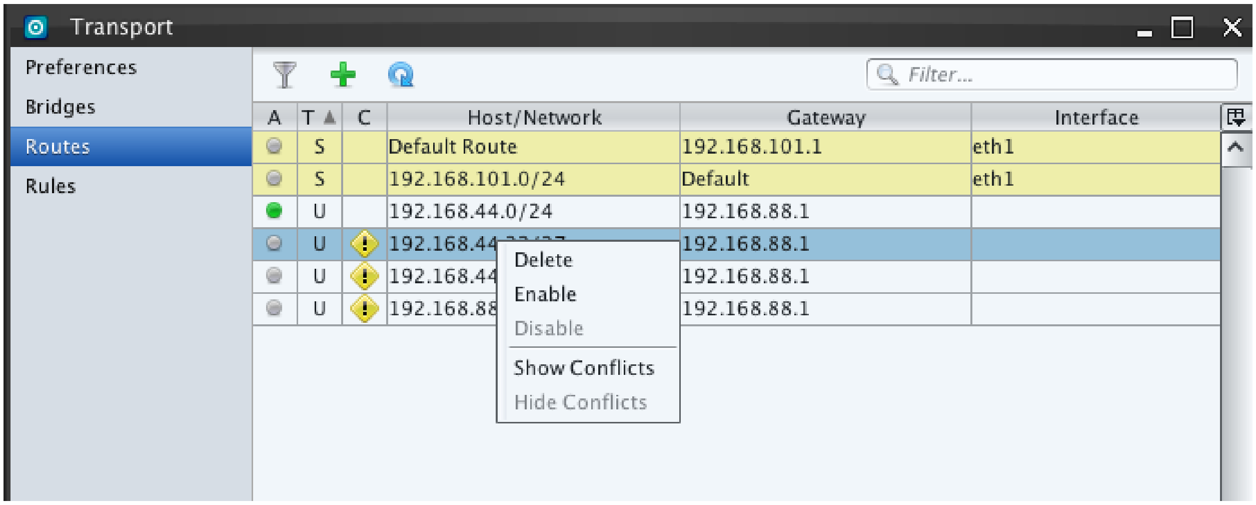
- Host or Network: Identifies the IP address for the destination host or the network address range. Packets matching the address range are redirected into the corresponding gateway.
- Gateway: Identifies the IP address for the gateway. If the gateway is the virtual interface's gateway *Rules* are evaluated.
- Interface: Identifies the interface for the route entry.

The first two rows in the Routes table are always highlighted in yellow and contain the default route and the local network's route. The route table has a right click menu to perform operations on selected route entries.

> - Delete: Removes a route entry from the table. Available only to inactive routes (including system routes).
> - Enable: Activates a User route entry by adding it to the operating system's route table.
> - Disable: Disables the route entry by removing the corresponding route from the operating system's route table. Available only to active User routes.
> - Show / Hide Conflicts: Enters and exits the Routes table's *Conflict Resolution Mode*.

Route Type Filtering

Systems with a large number of routes may overwhelm in the Routes table. Operators can filter route types out of the Routes table by checking or unchecking the appropriate checkbox in *Preferences Panel*. The current route filtering status displays in the center of the RoutePanel's status bar. If the status bar says *Type: All* then all route types are



A	T ▲	C	Host/Network	Gateway	Interface
●	S		Default Route	192.168.101.1	eth1
●	S		192.168.101.0/24	Default	eth1
●	U		192.168.44.0/24	192.168.88.1	
●	U	⚠	192.168.44	192.168.88.1	
●	U	⚠	192.168.44	192.168.88.1	
●	U	⚠	192.168.88	192.168.88.1	

Fig. 10.11: **Route Menu**

currently displayed. If the status bar says *Type: Minimal* then only the routes in conflict, the default route, and gateway entries are shown. Routes in in conflict ❗ are always shown regardless of the filter settings.

Routing State

The routing state is indicated located at the right side of the status bar, which includes:

- Routes: Total number of routes.
- State: Routing state.

State	Description
No Traffic	No traffic will be redirected.
Partial Traffic	Active routes are being redirected to their specified gateway.
Full Traffic	All traffic will be redirected to the virtual interface for *Rules* evaluation.

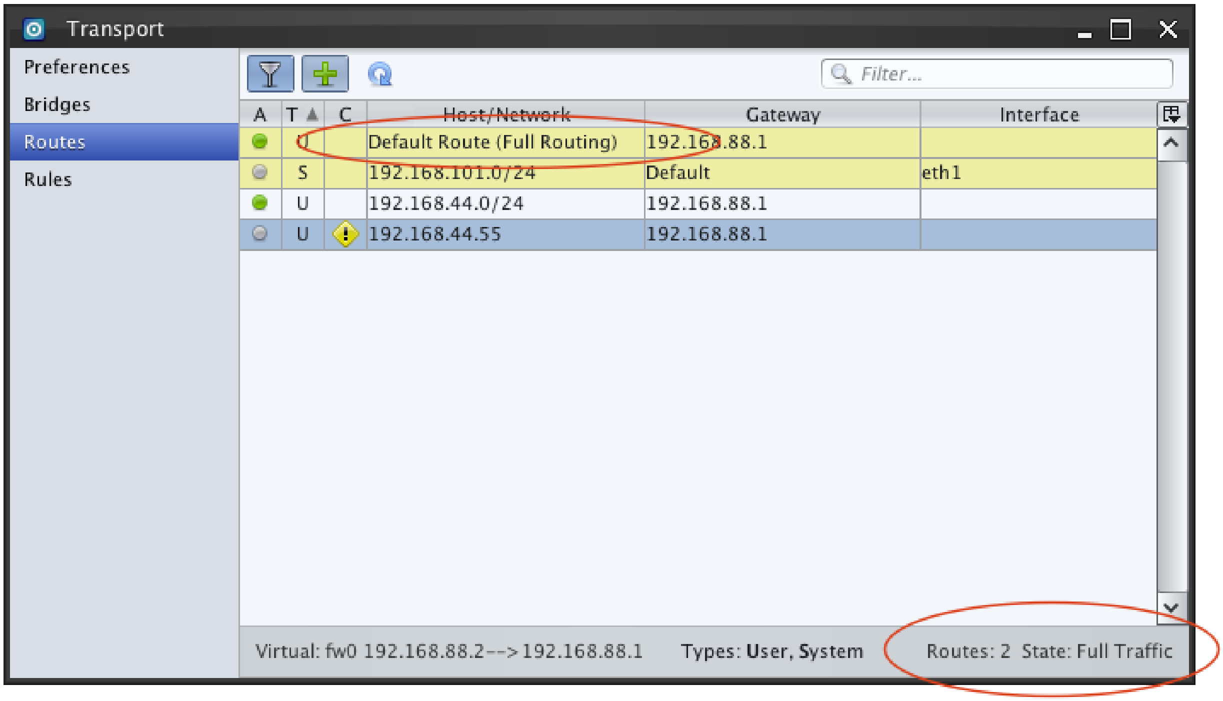
Full Routing

The "Full Routing" button toggles full routing on and off. Full routing redirects all packets that originate from the operator's workstation and relays them to the virtual gateway device. Full routing allocates a route entry with a wildcard host and network. Operators can disable full routing by toggling the button off or by manually disabling the full routing entry in the route table.

Partial Routing

Partial routing redirects packets that match active User routes defined in the route table. Unlike full routing, partial routing allows operators to selectively route traffic across the mesh network.

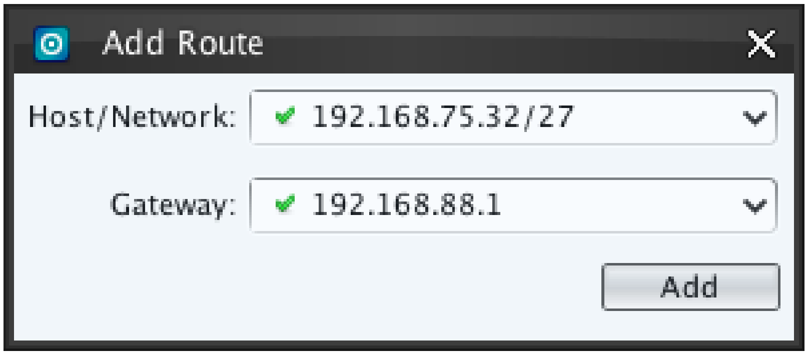
➕ Using the "Add Route" button, an operator can add an inactive route.



A	T	C	Host/Network	Gateway	Interface
●	U		Default Route (Full Routing)	192.168.88.1	
○	S		192.168.101.0/24	Default	eth1
●	U		192.168.44.0/24	192.168.88.1	
○	U	⚠	192.168.44.55	192.168.88.1	

Virtual: fw0 192.168.88.2-->192.168.88.1 Types: User, System Routes: 2 State: Full Traffic

Fig. 10.12: **Full Routing**



Host/Network: ✓ 192.168.75.32/27

Gateway: ✓ 192.168.88.1

Fig. 10.13: **Add Route**

- Host or Network: The network address to be redirected in the format: *xxx.xxx.xxx.xxx/mask*.
- Gateway: The IP address of the gateway. The default is to redirect packets into the virtual interface's gateway.

User routes are added in a disabled state. Once User routes are successfully added to the route table, the operator can activate partial routing by enabling one or more User routes. Full routing must be off. The *Routing State* updates in the status bar to indicate partial routing.

Query System Routes

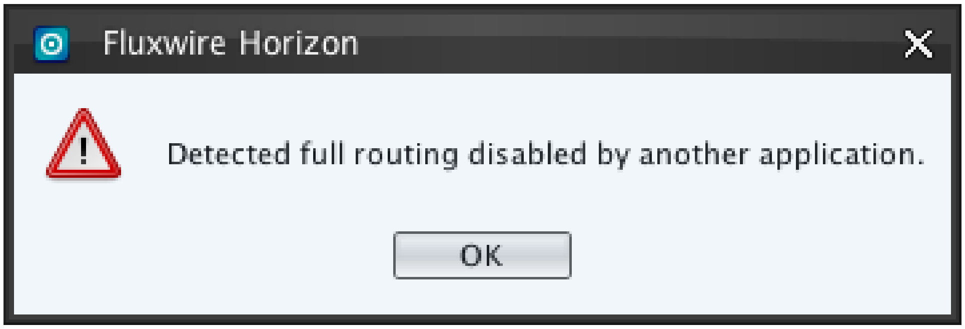
The routes in the Local Control's operating system's route tables may conflict with User routes. Fluxwire can perform polling of the operating system's route tables and provide the Routes table with all of these entries. If operators on the Local Control (or other software on that host) periodically alter system routes, operators should querying system routes more frequently. It is a good idea to refresh system routes before enabling new User routes to identify potential conflicts.

↻ Pressing the "Refresh System Routes" button causes all system routes to be pulled from the Local Control's operating system's routing tables.

Note: Querying system routes may make Horizon unresponsive for several seconds. This is because Horizon is checking for routing conflicts. Refreshing 2000 system routes can make Horizon unresponsive for up to a minute.

Route Auditing

The Local Control periodically checks the operating system's routing tables ever two seconds to detect if User or Link routes have been removed. The primary purpose of this feature is to alert Horizon if another application or system component has removed full routing mode's default route. When the full routing's default route is replaced the following modal dialog will appear in Horizon to alert the user. This feature does **not** provide updated information on System routes. System route updates must be manually queried using the ↻ button.



Detected full routing disabled by another application.

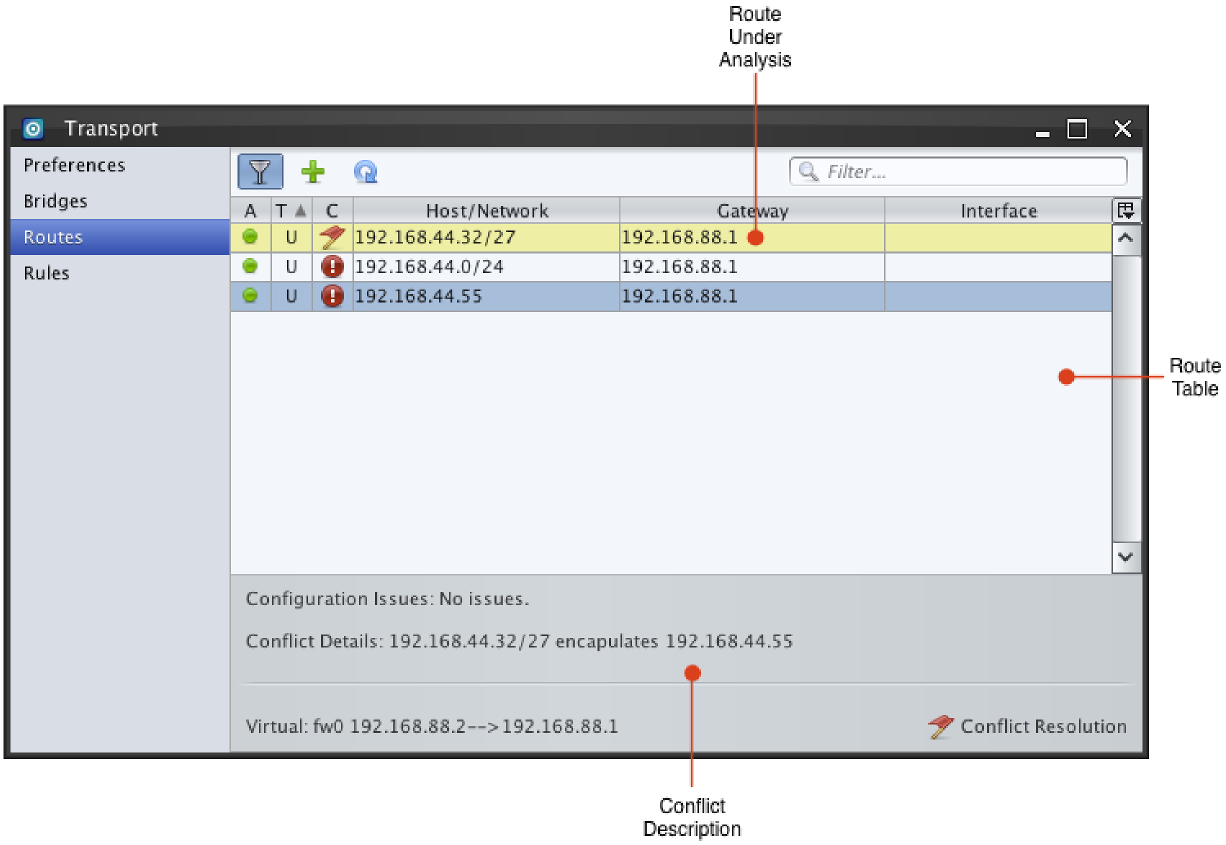
Fig. 10.14: **Default Route Removed**

Conflict Resolution Mode

The Route Panel can enter a special mode to resolve routing conflicts by clicking on "Show Conflicts" menu item for a route marked in conflict by ⚠ or ❗. Operators can choose to resolve conflicts or ignore them at their own discretion.

Once in this mode, the route under analysis is pinned at the top of the Routes table with a yellow background and a flag 🚩 in the Conflict (C) column. Details of conflicts are displayed directly below the Routes table in the Conflict Status Area. The Route Panel displays Fluxwire's virtual network configuration below the horizontal separator bar to assist in resolving route configuration issues. Selecting another route in the table reveals an explanation of why that particular route is in conflict with the route under analysis.

Note: An inactive User route marked with ⚠ does not need conflict resolution until the operator plans on enabling it.



A	T ▲	C	Host/Network	Gateway	Interface
●	U	🚩	192.168.44.32/27	192.168.88.1	
●	U	!	192.168.44.0/24	192.168.88.1	
●	U	!	192.168.44.55	192.168.88.1	

Configuration Issues: No issues.

Conflict Details: 192.168.44.32/27 encapulates 192.168.44.55

Virtual: fw0 192.168.88.2-->192.168.88.1

Fig. 10.15: **Route Conflict Resolution Mode**

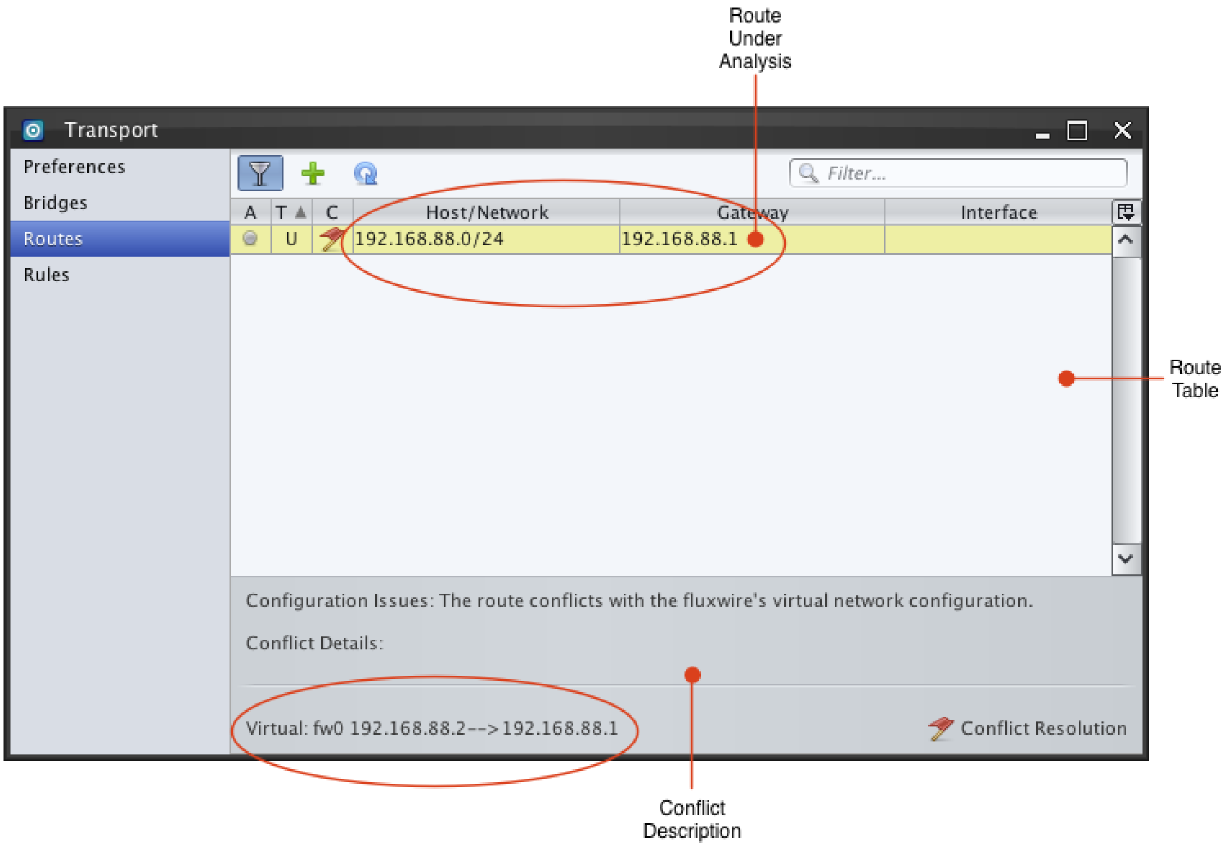
To resolve conflicts operators can right click on routes and click the "Disable" or "Delete" option. Operators can choose to view conflicts of another route by right clicking on the route and selecting the "Show Conflicts" option.

Configuration Issues for the route under analysis displays in the status area. Routes with configuration issues require either Transport's network configuration settings to be modified in the flx-desktop configuration file or the route under analysis to be deleted and recreated with appropriate changes.

Operators can leave Conflict Resolution Mode by right clicking and selecting the "Hide Conflicts" menu item. Deleting the route under analysis also causes the Route Panel to exit Conflict Resolution Mode.

Operators can leave routes in conflict at their discretion. One such example would be for an inactive route marked with ⚠. In this case, operators may be plan to swap out conflicting routes, keeping only one active at a time.

Note: Conflict Resolution Mode only shows active routes or system routes that the route under analysis



A	T ▲	C	Host/Network	Gateway	Interface
●	U		192.168.88.0/24	192.168.88.1	

Configuration Issues: The route conflicts with the fluxwire's virtual network configuration.

Conflict Details:

Virtual: fw0 192.168.88.2-->192.168.88.1

Conflict Resolution

Fig. 10.16: **Route with Gateway Configuration Issue**

conflicts with. Inactive routes may actually conflict with the route under analysis but will not be shown. Operators should resolve conflicts prior to enabling an inactive route marked with ⚠.

10.2.4 Rules

Rule evaluation determines which target Node acts as the gateway for new network sessions. Rules are evaluated once connections are routed to the Fluxwire virtual interface's gateway by Routes table entries. The first successfully evaluated rule determines if the connection is dropped or identifies a Node to for the connection to exit Fluxwire's mesh network from. Operators can select a default gateway Node and policy based on where they want their sessions to exit the mesh network. The default gateway Node can be set by right clicking on a Node in the Network Window and selecting "Gateway" in the Transport menu. The default rule's policy changes from **DROP** to **ALLOW** when *Full Routing* is set in the Routes panel.

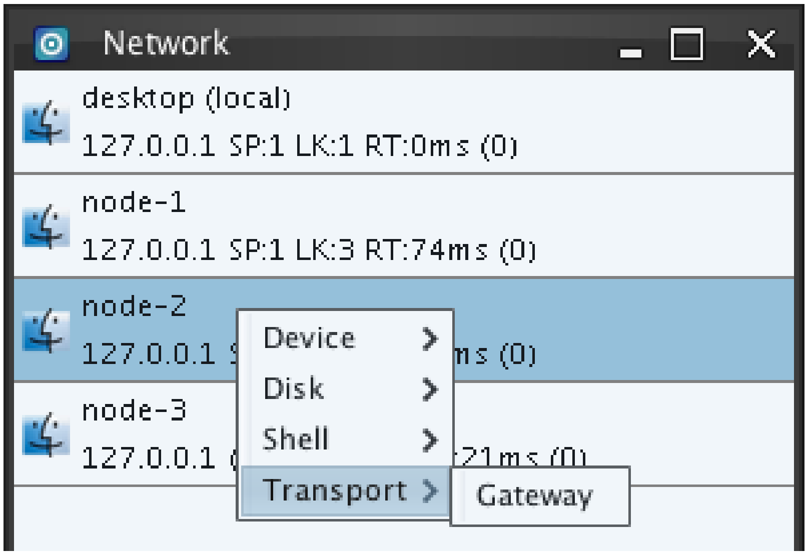


Fig. 10.17: **Gateway**

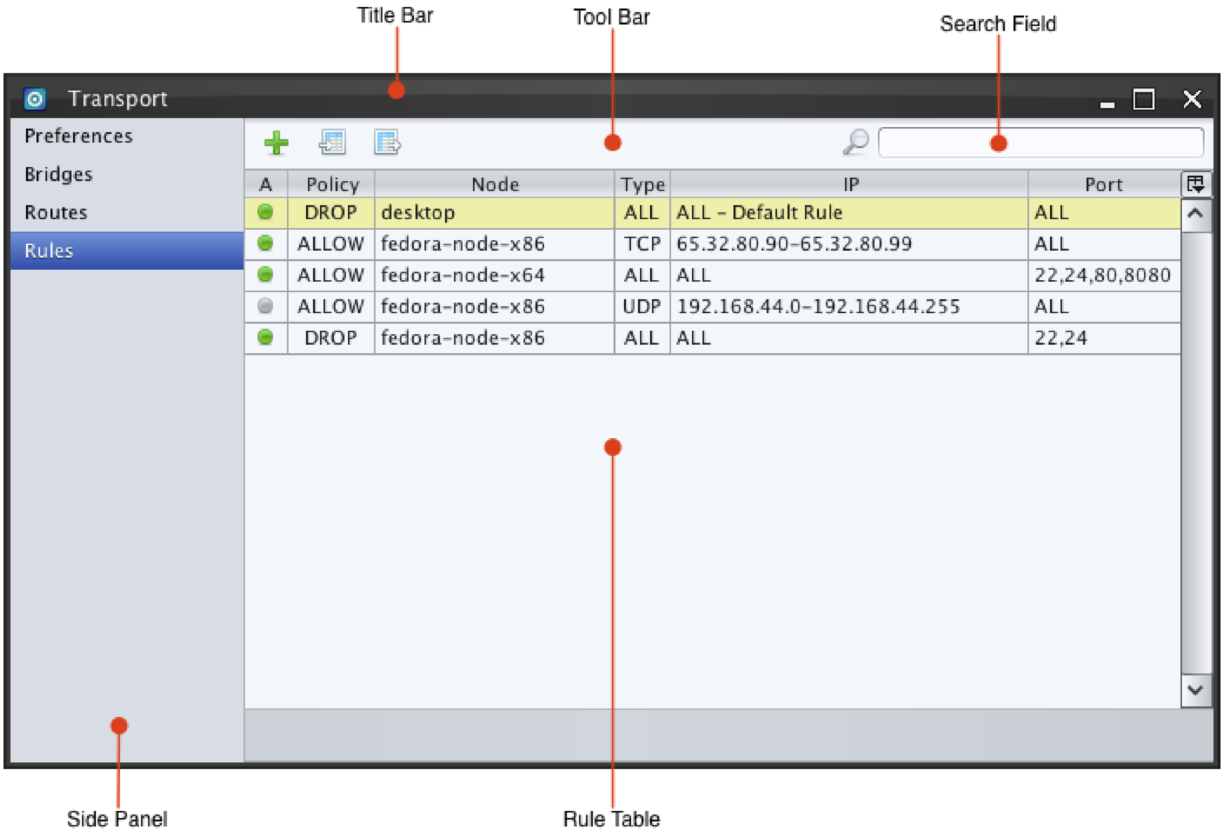
When an operator selects a new Gateway the Transport window is opened. This allows the operator to view all the transport rules. A default rule always exists and is marked with "Default Rule" in its IP column. The default rule is always pinned at the top of the Rules panel and highlighted in yellow.

The Rules table contains the following six columns.

- A: Active Status Icons

Symbol	Meaning
🟢	Rule is enabled and will be evaluated by Local Control.
⚪	Rule is disabled and will not be evaluated by Local Control.
⚠	Target Node is offline. Rule will not be evaluated by Local Control.

- Policy: **ALLOW** sessions matching the rule's criteria or **DROP** the sessions.



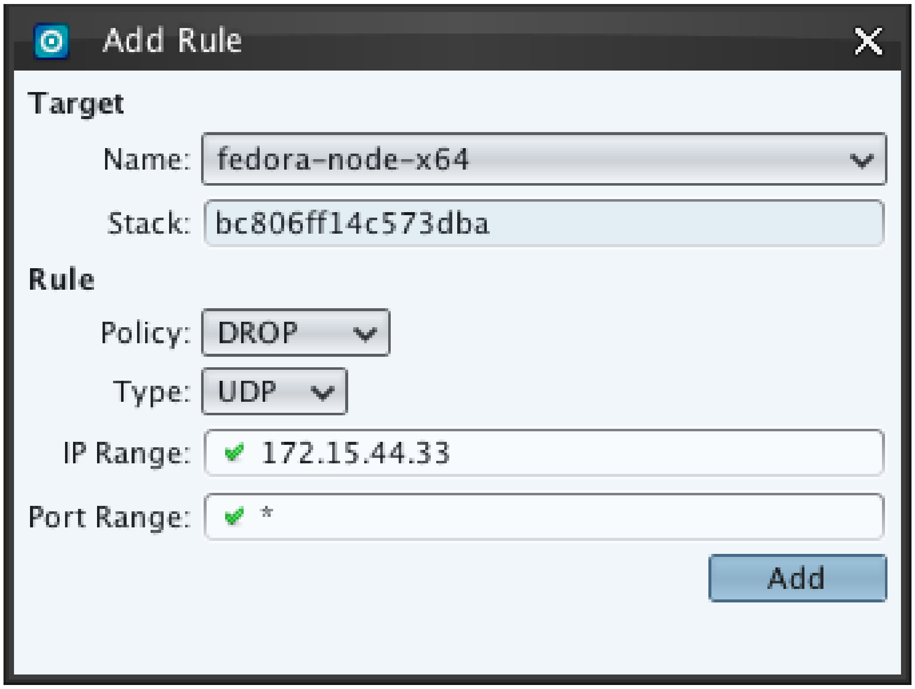
A	Policy	Node	Type	IP	Port
●	DROP	desktop	ALL	ALL – Default Rule	ALL
●	ALLOW	fedora-node-x86	TCP	65.32.80.90-65.32.80.99	ALL
●	ALLOW	fedora-node-x64	ALL	ALL	22,24,80,8080
○	ALLOW	fedora-node-x86	UDP	192.168.44.0-192.168.44.255	ALL
●	DROP	fedora-node-x86	ALL	ALL	22,24

Fig. 10.18: **Rules Panel**

- Node: The Node that is set as the rule's target gateway.
- Type: Link types. Either **TCP**, **UDP**, or **ALL** (both TCP and UDP).
- IP: The IP address or address range to match.
- Port: The port(s) or a port range to match.

Adding and Editing Rules

A new Rule is added by clicking the + button. The Rules Dialog appears and the user can create new rule definitions.



Target

Name: fedora-node-x64

Stack: bc806ff14c573dba

Rule

Policy: DROP

Type: UDP

IP Range: ✓ 172.15.44.33

Port Range: ✓ *

Fig. 10.19: **Rules Dialog - Add Operation**

The target Node gateway is selected in a combo box along with the policy to allow or drop sessions. The rule's criteria are specified by selecting the Link Type in the combo box and filling out the IP and Port criteria.

The **IP Range** can be set to a single IP address, a range, or a wildcard. It is not possible to specify multiple ranges or addresses a new rule must be created for each. Valid examples of **IP Range** input include:

- 44.33.22.11
- 65.88.55.33-65.88.55.233
- 192.168.33.0/24
- * Wildcard to match all IP addresses.

The **Port Range** can be set to a single port, a comma separated port list, a port range, or a wildcard. Port ranges cannot be mixed with comma separated port listings. Valid examples of **Port Range** include:

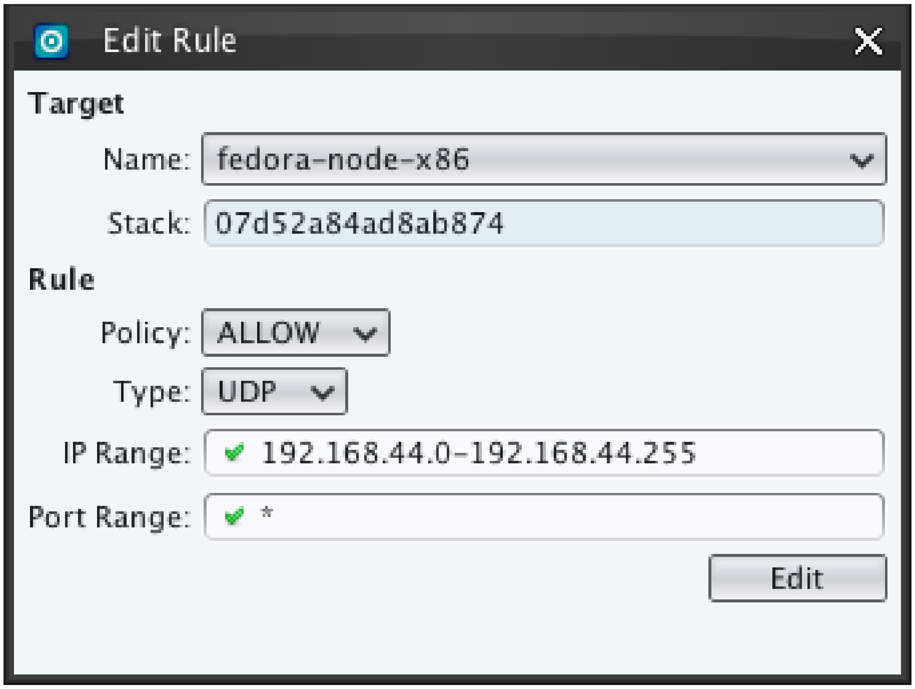
- 443
- 80,8080,443

- 501-550
- * Wildcard to match all ports.

Once valid criteria are set, a rule can be created by clicking the “Add” button. The dialog does not dismiss when creating new rules so that users can create multiple rules quickly. All rules default to the disabled ● status.

Rules can be enabled or disabled by right clicking and selecting “Enable” menu item. Disabled rules can be deleted by clicking the “Delete” menu item.

A disabled rule can be modified by right clicking and selecting “Edit” in the popup menu. The Rules Dialog will appear and be pre-populated with the selected rule. The “Add” button in the Rules Dialog is replaced by an “Edit” button.



Edit Rule

Target

Name: fedora-node-x86

Stack: 07d52a84ad8ab874

Rule

Policy: ALLOW

Type: UDP

IP Range: ✓ 192.168.44.0-192.168.44.255

Port Range: ✓ *

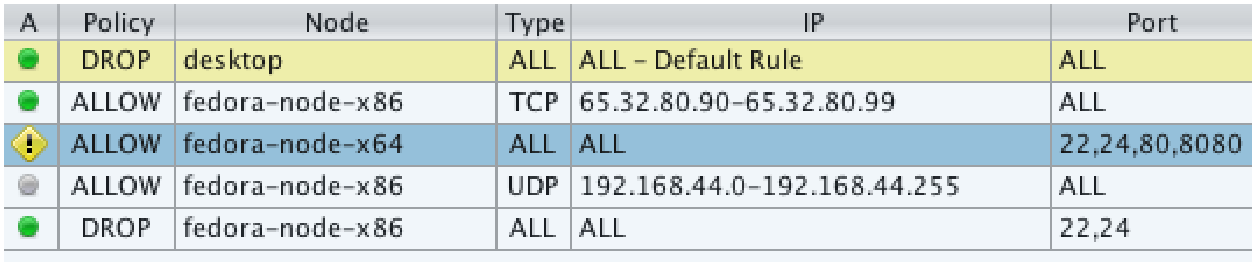
Fig. 10.20: **Rules Dialog - Edit Operation**

Rule Evaluation

The default rule always displays at the top of the rules table and contains the phrase “Default Rule” in the IP column. New rules are added at the bottom of the Rules table. The order of the rules in the table indicates the order the Local Control evaluates the rules. The policy and target gateway Node of the first matching rule is utilized. No other rules are evaluated after the first match. For example, the last entry in the Rule Table with a DROP policy will never get evaluated because ports 22,24 with IP range ALL will match the third row in the table.

Rules can be enabled and disabled. Only enabled ● rules are tested during rule evaluation. Rules are skipped when they are disabled ● or their target gateway Node is unreachable ⚠. When a Node is shutdown or their links disconnect or time out from Fluxwire’s mesh network all rules targeting the offline Node update their status to ⚠.

Warning: If the Node selected as the default rule’s gateway goes down the default rule reselects the Local Control and a DROP sessions policy.



A	Policy	Node	Type	IP	Port
🟢	DROP	desktop	ALL	ALL – Default Rule	ALL
🟢	ALLOW	fedora-node-x86	TCP	65.32.80.90-65.32.80.99	ALL
⚠	ALLOW	fedora-node-x64	ALL	ALL	22,24,80,8080
⚪	ALLOW	fedora-node-x86	UDP	192.168.44.0-192.168.44.255	ALL
🟢	DROP	fedora-node-x86	ALL	ALL	22,24

Fig. 10.21: **Rules Table: Selected row's Node lost connectivity**

Rules can be dragged and dropped in the Rules table with the mouse to change their evaluation order on the Local Control. Dragging a rule higher in the list causes it to be tested sooner and vice versa. Multiple rules can be selected and dragged together. If the selected rules were not already consecutive before the drag operation they will be packed consecutively when dropped.

Note: Rules are evaluated only when new network sessions originate. Existing connections do not reroute after rules are modified.

Note: Full or partial routing must be enabled for any rules to evaluate.

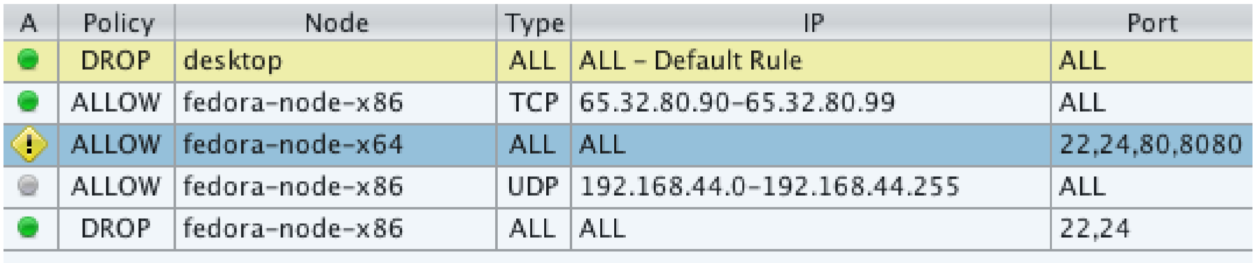
Importing and Exporting Rules

Horizon allows the current set of rules in the Rule Panel to be exported to a semicolon-delimited text file by pressing the "Export Rules" button. Entries flow left to right in the same order as they are shown in the Rules panel:

```
DISABLED;ALLOW;TCP;07d52a84ad8ab874;42f73913e3981e1f9a35781748b91425;65.32.80.90-65.32.80.99;*
```

Asterisks * indicate wildcard for either the IP range or the port range. In the example above port range has a wildcard.

Exported rule files can be imported on any Local Control by pressing the "Import Rules" button. If a rule gets discarded due to a parsing error a log entry is written to the Log Window and the Horizon's log file. Imported rules will import in the same ordering as they appear in the delimited rules file and be grouped together below all preexisting rules in the rules table. If the target Node cannot be found the ⚠ status is displayed and the hexadecimal stack identifier is printed in place of the Node's name.



A	Policy	Node	Type	IP	Port
🟢	DROP	desktop	ALL	ALL – Default Rule	ALL
🟢	ALLOW	fedora-node-x86	TCP	65.32.80.90-65.32.80.99	ALL
⚠	ALLOW	fedora-node-x64	ALL	ALL	22,24,80,8080
⚪	ALLOW	fedora-node-x86	UDP	192.168.44.0-192.168.44.255	ALL
🟢	DROP	fedora-node-x86	ALL	ALL	22,24

Fig. 10.22: **Rules Table: Unknown offline Node is target**

Note: Imported rules always set as disabled for safety.

10.3 Sessions

The Session table displays real-time table format of the VPN metadata. The Session Window is accessible by right clicking on the Local Control in the Network Window and selecting the Session option in the Transport menu. The Sessions table shows recent TCP and UDP connections from the Local Control that have successfully routed through the virtual interface's gateway.

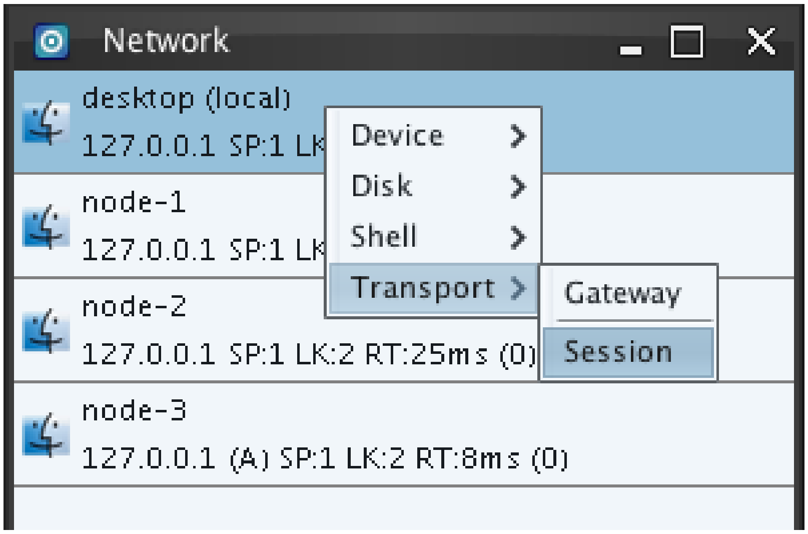


Fig. 10.23: **Sessions Menu Option**

Using the "Refresh Session" button, the operator can request a refresh of all active sessions. The table will be populated with the latest active sessions.

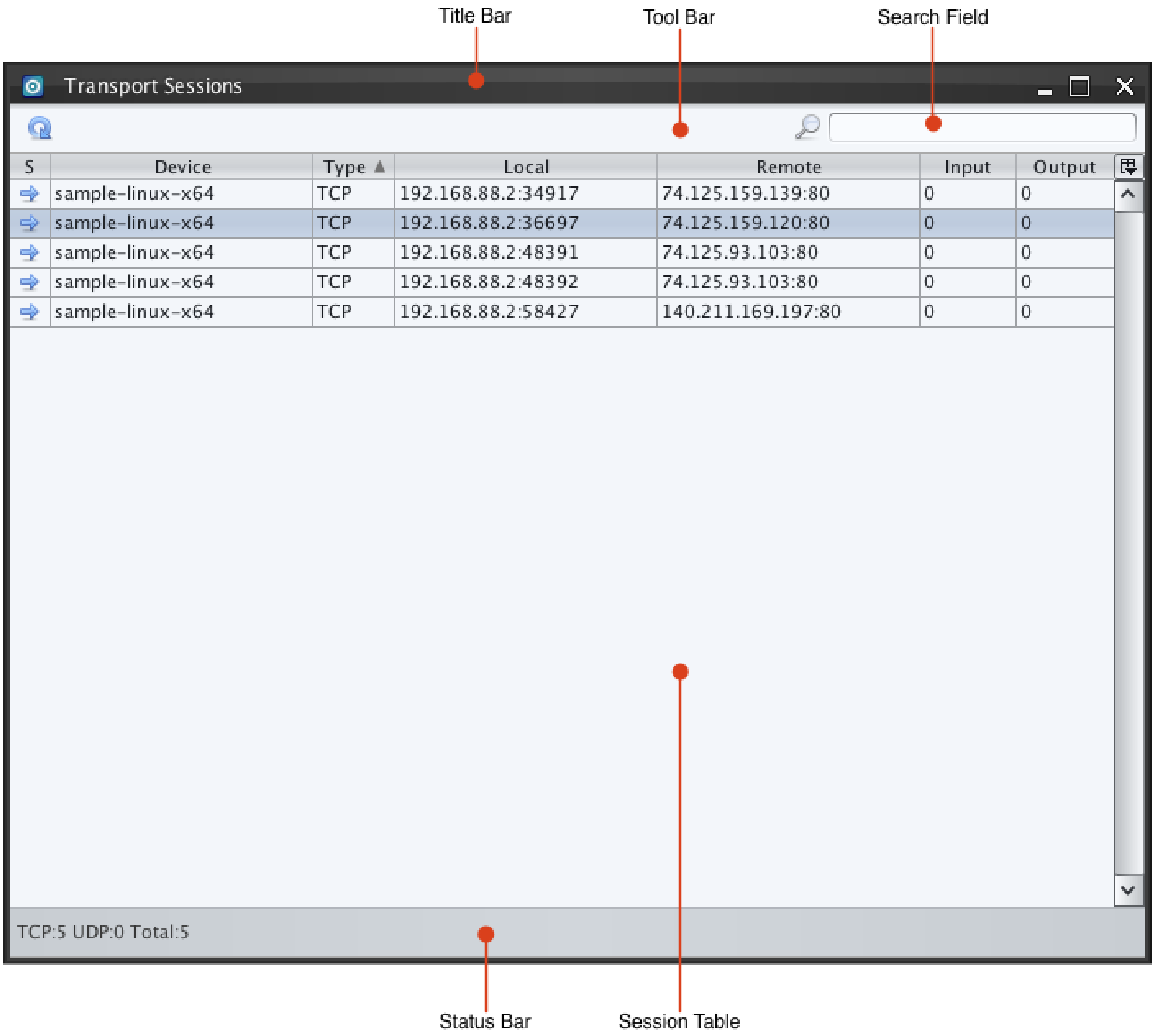
The session table is organized into six columns that describe the following:

- S (State): Indicates the state of the session; inbound or outbound connection using left arrow or right arrow respectively.
- Device: Name of the remote device for this session.
- Local: Local IP address and port for this session.
- Remote: Remote IP address and port for this session.
- Input: Total number of inbound bytes (Not Available).
- Output: Total number of outbound bytes (Not Available).

10.4 Dashboard Module Status

The Transport Module Status Panel in the Dashboard provides an operator with the current gateway, the number of active bridges, the number of active routes, the number of active TCP/UDP connections, and Accept/Drop counts.

TCP/UDP connection counts will vary over time depending on the activity or actions being performed, as shown in the following figures. However, there is a limit to the number of active TCP connections allowed, 2408. Thresholds (low, moderate, high) have been setup in the GUI to alert an operator of their current state. A low number of connections, <1200, is indicated by a TCP count with no highlighting. A moderate number of connections, 1200<= x <1800, is

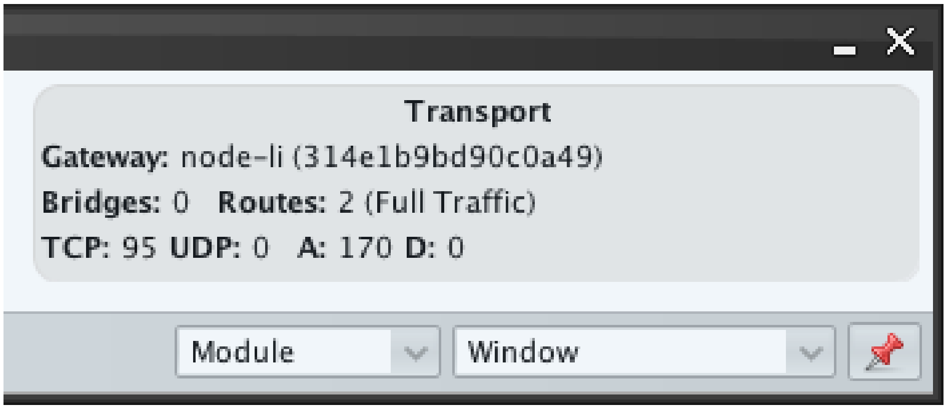


S	Device	Type ▲	Local	Remote	Input	Output
➡	sample-linux-x64	TCP	192.168.88.2:34917	74.125.159.139:80	0	0
➡	sample-linux-x64	TCP	192.168.88.2:36697	74.125.159.120:80	0	0
➡	sample-linux-x64	TCP	192.168.88.2:48391	74.125.93.103:80	0	0
➡	sample-linux-x64	TCP	192.168.88.2:48392	74.125.93.103:80	0	0
➡	sample-linux-x64	TCP	192.168.88.2:58427	140.211.169.197:80	0	0

TCP:5 UDP:0 Total:5

Fig. 10.24: **Sessions Window**

indicated by a TCP count highlighted in **yellow**. A high number of connections, 1800<= x <=2408, is indicated by TCP count highlighted in **red**.



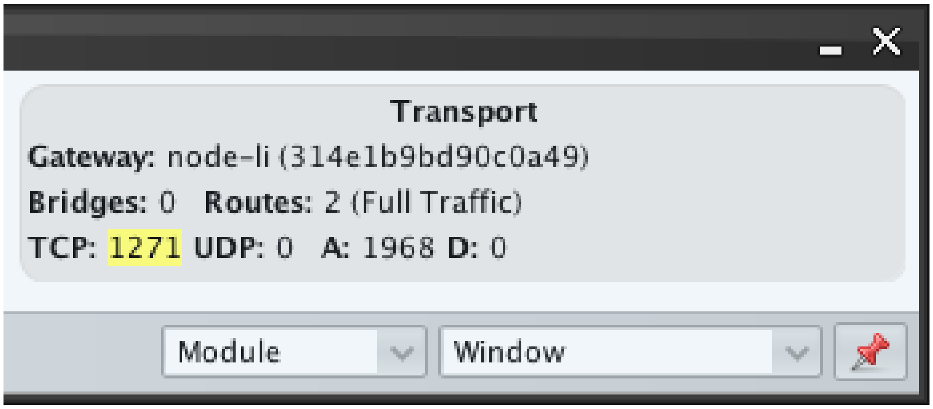
Transport

Gateway: node-li (314e1b9bd90c0a49)

Bridges: 0 **Routes:** 2 (Full Traffic)

TCP: 95 **UDP:** 0 **A:** 170 **D:** 0

Fig. 10.25: **Transport Status Panel - TCP Connection Count Low**



Transport

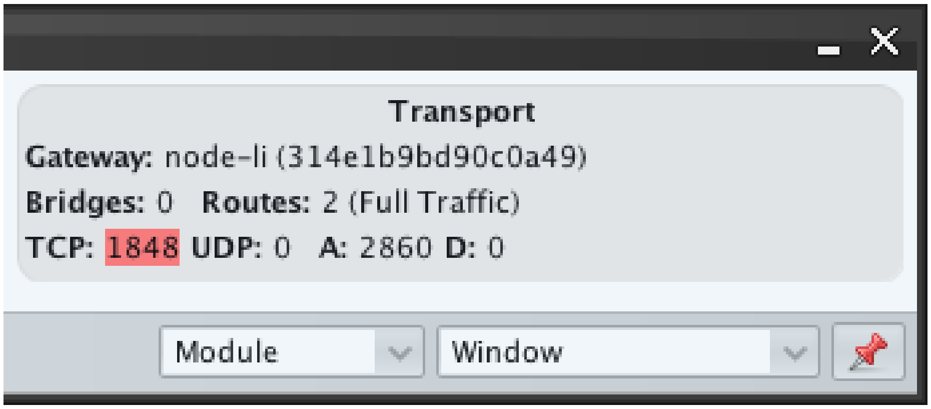
Gateway: node-li (314e1b9bd90c0a49)

Bridges: 0 **Routes:** 2 (Full Traffic)

TCP: 1271 **UDP:** 0 **A:** 1968 **D:** 0

Fig. 10.26: **Transport Status Panel - TCP Connection Count Moderate**

Warning: Applications such as nmap will at times create a high TCP connection count. If the count reaches the maximum allowed connections, 2048, and remains there results from nmap may be inconsistent or wrong. To prevent this, scans should be carefully crafted and limited to one scan at a time.



Transport

Gateway: node-li (314e1b9bd90c0a49)

Bridges: 0 Routes: 2 (Full Traffic)

TCP: 1848 UDP: 0 A: 2860 D: 0

Fig. 10.27: **Transport Status Panel - TCP Connection Count High**

CHAPTER

ELEVEN

API

The tools package contains a python library, called the Control-API, that provide a programmatic interface for client-server interactions. Operators can use this library to build tools and scripts that automate, simplify, and streamline complex sets of actions and events. The interface is composed of multiple python classes that allow operators to issue commands to both Controls and Nodes as well all available plugins. For more details, please refer to the Control-API documentation bundled with the tools package.

11.1 Graph

Additionally, Control-API supports drawing network graphs to many different formats, such as DOG, SVG, VDX (Visio). The graphs are constructed using NetworkX and rendered with Graphviz. Many custom options are available for layout, color, and labeling. An example SVG graph is shown below:

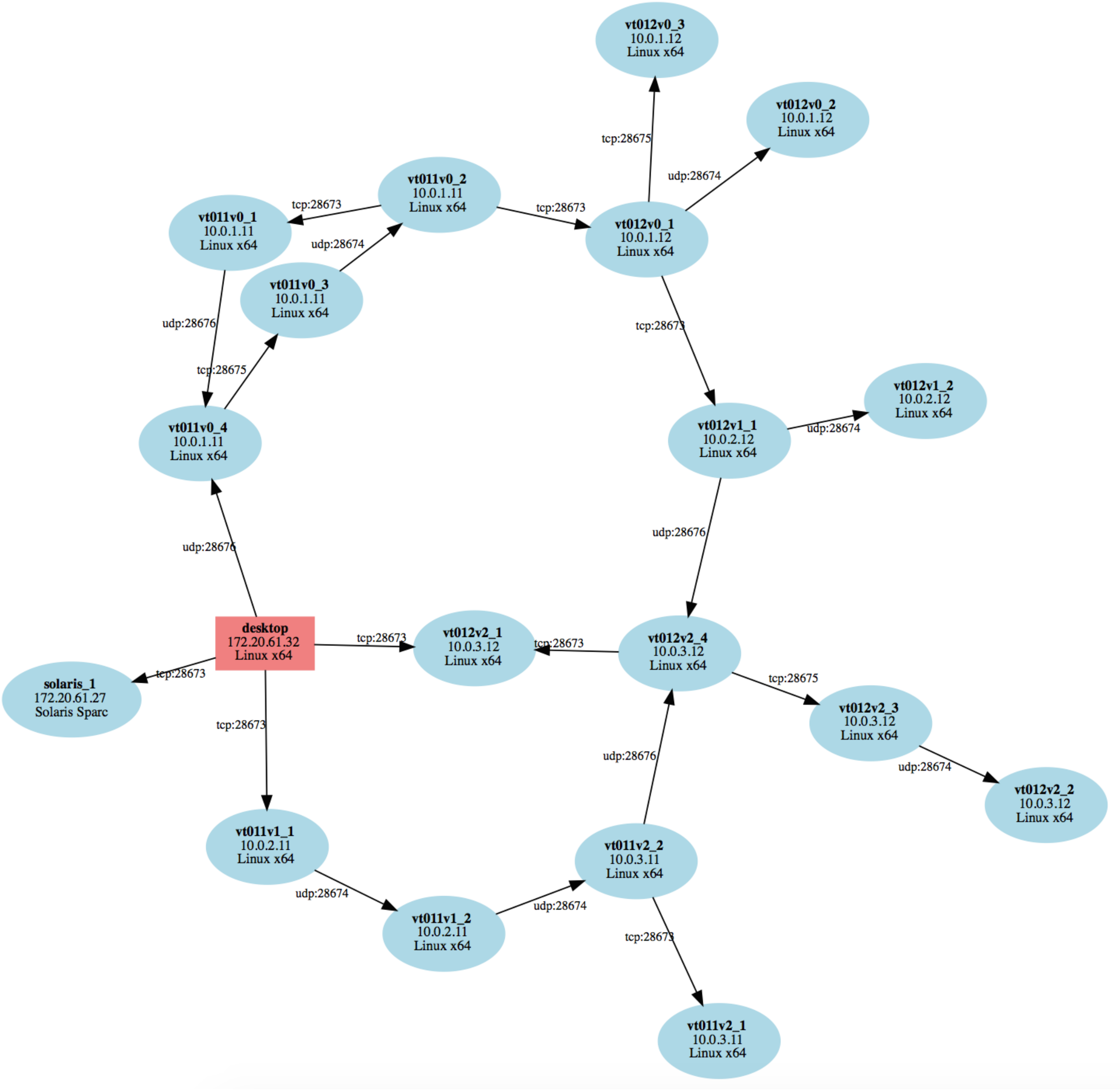


Fig. 11.1: **SVG Network Graph**

CHAPTER

TWELVE

DATABASES

12.1 Device

12.1.1 Schema

```
CREATE TABLE IF NOT EXISTS uid (
       pk            INTEGER PRIMARY KEY,
       hash          TEXT,
       hardware_id   BLOB,
       UNIQUE(hardware_id)
);
```

```
CREATE TABLE IF NOT EXISTS devices (
       pk             INTEGER PRIMARY KEY,
       device_id      TEXT,
       stack_id       TEXT,
       hardware_pk    INTEGER,
       pid            INTEGER,
       service_count  INTEGER DEFAULT 0,
       link_count     INTEGER DEFAULT 0,
       plugin_count   INTEGER,
       plugins        TEXT,
       type           TEXT,
       name           TEXT,
       hostname       TEXT,
       domain         TEXT,
       os             TEXT,
       cpu            TEXT,
       model          TEXT,
       version        TEXT,
       locale         TEXT,
       tz_minuteswest INTEGER,
       tz_dsttime     INTEGER,
       timeout        INTEGER,
       interval       INTEGER,
       lastupdate     DATETIME DEFAULT CURRENT_TIMESTAMP,
       offline_sync   INTEGER DEFAULT 0,
       UNIQUE(device_id)
);
```

```
CREATE TABLE IF NOT EXISTS links (
       pk                 INTEGER PRIMARY KEY,
       device_pk          INTEGER,
       state              INTEGER,
       id                 INTEGER,
       event              INTEGER,
       mtu                INTEGER,
       overhead           INTEGER,
       address            TEXT,
       type               TEXT,
       proto              TEXT,
       ehost              TEXT,
       eport              INTEGER,
       lhost              TEXT,
       lport              INTEGER,
       rhost              TEXT,
       rport              INTEGER,
       lastupdate         DATETIME DEFAULT CURRENT_TIMESTAMP,
       watchdog_timeout   INTEGER,
       watchdog_retries   INTEGER,
       UNIQUE(device_pk, type, proto, eport, ehost, lport, lhost, rport, rhost),
       FOREIGN KEY(device_pk) REFERENCES devices(pk)
);
```

```
CREATE TABLE IF NOT EXISTS services (
       pk          INTEGER PRIMARY KEY,
       device_pk   INTEGER,
       state       INTEGER,
       id          INTEGER,
       type        TEXT,
       proto       TEXT,
       port        INTEGER,
       host        TEXT,
       lastupdate  DATETIME DEFAULT CURRENT_TIMESTAMP,
       UNIQUE(device_pk, type, proto, port, host),
       FOREIGN KEY(device_pk) REFERENCES devices(pk)
);
```

12.1.2 Details

Table 12.1: UID

Column	Description
pk	Primary Key
hash	UID hash - hex format
hardware_id	Unique hardware ID blob

Table 12.2: Devices

Column	Description
pk	Primary Key
device_id	Device ID
stack_id	Stack ID
hardware_pk	UID primary key (reference to UID row)
pid	Process ID
service_count	Number of services
link_count	Number of links
plugin_count	Number of plugins
plugins	List of loaded plugins
type	Type: control/node
name	User supplied name of control/node
hostname	Host name
domain	Domain name, if available
os	Operating system (linux, mswin, . . .)
cpu	CPU type (x86, x64, . . .)
model	System model (mikrotik, paradyne, . . .)
version	System version
locale	Language and character set
tz_minuteswest	Number of minutes west of UTC
tz_dsttime	Daylight savings time, enabled/disabled
timeout	Alarm timeout
interval	Alarm interval
lastupdate	Last update timestamp
offline_sync	Marker to indicate if device is still active in the offline list

Table 12.3: Links

Column	Description
pk	Primary Key
device_pk	Device primary key (reference to Devices row)
state	State of link
id	Link ID
event	Event ID
mtu	Link MTU
overhead	Overhead value
address	Stack address
type	Type: TCP/UDP
proto	Protocol: None/HTTP
eport	External port
ehost	External host
lport	Local port
lhost	Local host
rport	Remote port
rhost	Remote host
lastupdate	Last update timestamp
watchdog_timeout	Timeout in seconds
watchdog_retries	Number of retries

Table 12.4: Services

Column	Description
pk	Primary Key
device_pk	Device primary key (reference to Devices row)
state	State
id	Service ID
type	Type: TCP/UDP
proto	Protocol: None/HTTP
port	Port number
host	Host address
lastupdate	Last update timestamp

12.2 Disk

12.2.1 Schema

```
CREATE TABLE IF NOT EXISTS devicecache (
       pk       INTEGER PRIMARY KEY,
       hash     TEXT NOT NULL,
       stack_id TEXT NOT NULL
);
```

```
CREATE TABLE IF NOT EXISTS filesystem (
       pk             INTEGER PRIMARY KEY,
       devicecache_pk INTEGER,
       path           TEXT DEFAULT "",
       filename       TEXT DEFAULT "",
       uid            INTEGER DEFAULT -1,
       gid            INTEGER DEFAULT -1,
       mode           INTEGER DEFAULT 0,
       size           INTEGER DEFAULT -1,
       atime          INTEGER DEFAULT 0,
       mtime          INTEGER DEFAULT 0,
       ctime          INTEGER DEFAULT 0,
       valid          INTEGER DEFAULT 2,
       err            TEXT DEFAULT "",
       linktarget     TEXT DEFAULT NULL,
       linktargetmode INTEGER DEFAULT 0,
       linktype       INTEGERDEFAULT0,
       lastupdate     DATETIME DEFAULT CURRENT_TIMESTAMP,
       hash           TEXT DEFAULT "",
       UNIQUE(devicecache_pk,path,filename)
);
CREATE INDEX IF NOT EXISTS idxsourcepathfilename ON filesystem (source,path,filename);
CREATE INDEX IF NOT EXISTS idxsourcepathvalid ON filesystem (source,path,valid);
```

```
CREATE TABLE IF NOT EXISTS transfers(
       pk             INTEGER PRIMARY KEY,
       up_id          INTEGER,
       up_node        TEXT,
       up_file        TEXT,
       down_id        INTEGER,
       down_node      TEXT,
       down_file      TEXT,
       down_tmp_file  TEXT,
       state          INTEGER,
       registered     TEXT,
       persist        TEXT,
       size           INTEGER,
       bytes_sent     INTEGER,
       status         TEXT,
       lastupdate     DATETIME DEFAULT CURRENT_TIMESTAMP,
       UNIQUE(up_id, up_node, down_id, down_node)
);
CREATE INDEX IF NOT EXISTS idxupdown ON transfers (up_id, up_node, down_id);
```

12.2.2 Details

Table 12.5: Devicecache

Column	Description
pk	Primary Key
hash	UID hash - hex format
stack_id	Stack ID

Table 12.6: Filesystem

Column	Description
pk	Primary Key
devicecache_pk	Device cache primary key (reference to Devicecahe row)
path	Directory path to file
filename	Filename
uid	UID
gid	GID
mode	File mode (permissions)
size	Size in bytes
atime	Access time
mtime	Modified time
ctime	Create time
valid	Currently valid
err	Error flag
linktarget	Full path of a link's target file or directory
linktargetmode	File mode (permissions) of a link's target file or directory
linktargettype	Type of link
lastupdate	Last update timestamp
hash	MD5 hash

Table 12.7: Transfers

Column	Description
pk	Primary Key
up_id	Upload ID
up_node	Upload node stack address
up_file	Full path of upload file (path and filename)
down_id	Download ID
down_node	Download node stack address
down_file	Full path of download file (path and filename)
down_tmp_file	Full path of temporary download file (path and filename)
state	State of file transfer
registered	Registered flag option
persist	Persist flag option
size	Size of file being transferred
bytes_sent	Number of bytes transferred
status	Status of file transfer
lastupdate	Last update timestamp

12.3 Transport

12.3.1 Schema

```
CREATE TABLE IF NOT EXISTS network (
       status INTEGER,
       hash   BLOB,
       target BLOB,
       start  DATETIME DEFAULT CURRENT_TIMESTAMP,
       end    DATETIME DEFAULT CURRENT_TIMESTAMP,
       proto  INTEGER,
       lhost  TEXT,
       lport  INTEGER,
       rhost  TEXT,
       rport  INTEGER,
       input  INTEGER DEFAULT 0,
       output INTEGER DEFAULT 0
);
```

12.3.2 Details

Table 12.8: Network

Column	Description
status	Status of connection
hash	Session identifier
target	Target stack address
start	Start timestamp
end	End timestamp
proto	Protocol type
lhost	Local host
lport	Local port
rhost	Remote host
rport	Remote port
input	Input bytes
output	Output bytes

CHAPTER

THIRTEEN

CONFIGURATION FILE

The configuration file, linux-fluxwire.conf, contains all of the default values used by Fluxwire. These values are read in when flx-desktop is launched and stored for the duration of the application. Any changes made to linux-fluxwire.conf will require that flx-desktop be restarted so that it can read the new values in. Some values are passed to the Graphical User Interface when it connects allowing an operator to further refine the settings in real-time. The .conf file is broken into the following sections: Version, Desktop, Device, Transport, Chat, Shell, Loader, Disk, HTTP, and HTTPS.

13.1 Version

```
version = "3.4.1";
```

Note: If flx-desktop is launched with a .conf file that does not contain the same version an error will be displayed and flx-desktop will shutdown. This example shows a 3.4.0 .conf file being used with a 3.4.1 flx-desktop.

```
# flx-desktop linux-fluxwire.conf

    Fluxwire Desktop v3.4.1 importing config parameters:[e] .conf version mismatch:
        .conf:2.4.0 desktop:3.4.1 [ failed ]

desktop shutdown: [ failed ]
```

13.2 Desktop

```
desktop: {
  /* Name of the desktop instance */
  name = "desktop";

  directory: {
    /* Install directory */
    install = "/usr/local/fluxwire/default";

    /* Base directory for all logs and database storage for current session */
    session = "basedir";
  };

  clients: {
    host = "127.0.0.1";
    port = 8000;
    limit = 16;

    log: {
      level = 0xffffffff;
      storage = "auto";
    };
  };

  systems: {
    file = "systems.conf";
  };

  plugins: {
    /* Configuration file for all plugins provided by this deployment */
    file = "plugins.conf";

    /* Autoload settings for the local control and remote nodes/controls */
    autoload: {
      local = ( "transport", "disk", "shell", "chat", "loader" );
      remote = ( "transport" );
    };
  };
};
```

13.3 Device

```
device: {
  log : {
    /* Log level, see documentation for options */
    level = 0xffffffff;

    /* Log outputs include stdout, file, and network udp
     * file:stdout      --> redirect log to standard output
     * file:filename    --> redirect log to the filename
     * udp:address:port --> redirect log to address and port in udp
     */
    type = "auto";
  };

  stacks = ( { mtu = 1250; key = "aaaabbbbccccdddd";
               /* ipv = "4"; ipv = "6"; ipv = "dual"; "4" is default */
               services = ( { type = "tcp"; ipv="dual"; protocol = "none"; port = 443; },
                            { type = "udp"; ipv="dual"; protocol = "none"; port = 53; } );
             }
           );

  /* Determine if this control responds to timer requests from nodes. Can
   * be either on or off. Default value is always on even when this setting is
   * not explicitly set.
   */
  node_timer = "on";
};
```

13.4 Disk

```
disk: {
  log : {
    /* Log level, see documentation for options */
    level = 0x0000001f;

    /* Log outputs include stdout, file, and network udp
     * file:stdout      --> redirect log to standard output
     * file:filename    --> redirect log to the filename
     * udp:address:port --> redirect log to address and port in udp
     */
    type = "auto";
  };

  /* Number of times to overwrite a file before deleting it */
  overwrite_iterations = 7;

  /* Should directory resolve Windows links */
  resolve_win_links = 0;

  /* Automatically refresh the directory window after file operation */
  auto_refresh = 0;

  transfer: {
    /* Should file transfers persist if control goes down */
    persist = 1;

    /* Default download directory with available format options:
     *   %n  unique stack ID
     *   %N  operator defined node name in Horizon
     *   %t  current time
     *   %T  local module load time
     *   %p  full path from upload node
     */
    download_directory = "/root/fwdata-%T/%N-%n/%p/";

    /* Maximum number of active transfers */
    max_active = 0;

    directory: {
      /* Transfer view
       * file_count_stats_view - file count exceeds this value display stats instead
       *    of file tree. Value used to prevent 'Out of Memory' errors in Horizon.
       */
      file_count_stats_view = 10000;

      warning: {
        /* file_count - total number of files to be transferred exceeds this value
         * total_size - total amount of data exceeds this value (Mb)
         * last_update - last update is older than value (hours)
         */
        file_count = 100;
        total_size = 100;
        last_update = 12;
      };
    };
  };
};
```

13.5 Transport

```
transport: {
  log : {
    /* Log level, see documentation for options */
    level = 0xffffffff;

    /* Log outputs include stdout, file, and network udp
     * file:stdout      --> redirect log to standard output
     * file:filename    --> redirect log to the filename
     * udp:address:port --> redirect log to address and port in udp
     */
    type = "auto";
  };

  /* Virtual interface settings for all transport connections */
  virtual: {
    /* Interface name
     *  Linux : can be any name not used on the current interface list
     *  Darwin : must be tun0 through tun9
     *  Windows : must be similar to the name of the Tap interface under the Network
     *            Connections control panel.
     */
    interface = "fw0";

    /* Maximum transmission unit for the virtual interface, this should be set
     * a minimum 30 less than the mtu for the stack */
    mtu = 1200;

    /* Local ip address for the interface */
    address = "192.168.88.2";

    /* Remote ip address for the interface, must match the subnet of the local */
    gateway = "192.168.88.1";

    /* Subnet mask for the virtual interface network */
    netmask = "255.255.255.0";

    /* UDP timeout in seconds for idle connections */
    udp_timeout = 30;
  };

  /* System interface settings */
  physical: {
    /* Name of the main interface for the network */
    interface = "tun0";

    /* Local ip address for the network */
    address = "0.0.0.0";

    /* Gateway address of the main interface for the network */
    gateway = "0.0.0.0";

    /* Network subnet and netmask */
    network = "0.0.0.0/32";
  };
};
```

13.6 Chat

```
chat: {
  log : {
    /* Log level, see documentation for options */
    level = 0x0000001f;

    /* Log outputs include stdout, file, and network udp
     * file:stdout      --> redirect log to standard output
     * file:filename    --> redirect log to the filename
     * udp:address:port --> redirect log to address and port in udp
     */
    type = "auto";
  };
};
```

13.7 Loader

```
loader: {
  log : {
    /* Log level, see documentation for options */
    level = 0xffffffff;

    /* Log outputs include stdout, file, and network udp
     * file:stdout      --> redirect log to standard output
     * file:filename    --> redirect log to the filename
     * udp:address:port --> redirect log to address and port in udp
     */
    type = "auto";
  };
};
```

13.8 Shell

```
shell: {
  log : {
    /* Log level, see documentation for options */
    level = 0xffffffff;

    /* Log outputs include stdout, file, and network udp
     * file:stdout      --> redirect log to standard output
     * file:filename    --> redirect log to the filename
     * udp:address:port --> redirect log to address and port in udp
     */
    type = "auto";
  };
};
```

13.9 HTTP

```
http: {
  log : {
    /* Log level, see documentation for options */
    level = 0xffffffff;

    /* Log outputs include stdout, file, and network udp
     * file:stdout      --> redirect log to standard output
     * file:filename    --> redirect log to the filename
     * udp:address:port --> redirect log to address and port in udp
     */
    type = "auto";
  };
};
```

13.10 HTTPS

```
https: {
  log : {
    /* Log level, see documentation for options */
    level = 0xffffffff;

    /* Log outputs include stdout, file, and network udp
     * file:stdout      --> redirect log to standard output
     * file:filename    --> redirect log to the filename
     * udp:address:port --> redirect log to address and port in udp
     */
    type = "auto";
  };
};
```

CHAPTER

FOURTEEN

TUTORIALS

14.1 Build Nodes

Nodes are created by executing flx-packer with the appropriate flags and options. More information about *Packer (flx-packer)* and *Node* can be found in the *Applications* section. The sections below will describe each option in flx-packer while walking through the build process of a node. Each section will provide example(s) that will build upon the previous section's examples.

14.1.1 Authentication Code (-k)

All nodes require an Authentication Code which is used as a one-time pad key for handshakes between devices. The code must be an alphanumeric string sequence of 16 bytes. The authentication code is set with the **-k** option.

- Basic node with a Service Port.

```
$ flx-packer -o node-linux -s linux:x86 -k aaaabbbb12345678 \
  --service ``tcp=5001''
```

14.1.2 Device Name (-n)

Names can be assigned to each node through the **-n** option. Names are not required to be unique but using the same name for multiple nodes defeats the purpose of setting this option. Device names assist operators in identifying which device they are working with in the GUI rather than relying on the Stack Address (unique 16 byte hexadecimal value) of each device. Device Names can always be assigned/changed in the GUI as shown in *Identity Panel*.

- Linux node with name set.

```
$ flx-packer -o node-linux -s linux:x86 -k aaaabbbb12345678 \
  -n kung-foo --service ``tcp=5001''
```

- Microsoft Windows node with name set.

```
$ flx-packer -o node-linux -s mswin:x86 -k aaaabbbb12345678 \
  -n Bill_Gates --service ``tcp=5001''
```

14.1.3 Maximum Transmission Unit (-m)

A Maximum Transmission Unit can be set on a node by node basis and is used to define the size (in bytes) of the largest protocol data unit that the link can pass onwards (MTU). The value is set with the **-m** option and must be greater than or equal to 512 and it is recommended that the value not exceed 1500 for normal network operations.

Example(s)

- Node with a Maximum Transmission Unit value of 1300.

  ```
  $ flx-packer -o node-linux -s linux:x86 -k aaaabbbb12345678 \
    -n kung-foo -m 1300 --service ``tcp=5001''
  ```

14.1.4 Timeout (-i)

A timeout can be set which specifies when to shutdown and whether or not to delete the node when a timeout occurs. The timeout option consists of 3 components: timeout (seconds), interval (seconds), and a delete flag (on/off). A timeout occurs when no active users are on the network and no responses are received for any given interval within the timeout period. If a timeout does occur and the delete flag is set to *off* the node will be shutdown and the binary will remain on the file system. Otherwise the node will be deleted from the file system after shutdown. Timeouts can also be set in the GUI in the *Alarm Panel*. The format of the **-i** option is as follows:

```
-i <timeout>:<interval>:<cleanup on/off>
```

Example(s)

- Node with timeout of 1200 seconds, 30 second interval, and the delete flag set to *on*.

  ```
  $ flx-packer -o node-linux -s linux:x86 -k aaaabbbb12345678 \
    -n kung-foo -m 1300 -i 1200:30:on --service ``tcp=5001''
  ```

14.1.5 Service Ports (–service) and Links (–link)

Nodes can be configured to contain one or more service ports. This is done by adding one or more **–service** option flags. The service ports in each node only accepts incoming Fluxwire connections of the given type/protocol specified in the port configuration. The format of the **–service** option is as follows:

```
--service "<tcp/udp>=<port> proto=<protocol> <protocol configuration>"
```

Example(s):

- Host is connected directly to the Internet with no firewall or security software installed. TCP port 4000 and UDP port 4001 will be used.

  ```
  $ flx-packer -o node-linux -s linux:x86 -k aaaabbbb12345678 \
    -n kung-foo -m 1300 -i 1200:30:on --service ``tcp=4000'' --service ``udp=4001''
  ```

Operators can also setup default links in each node during the build process. Links and service ports can be combined as needed. One or more links can be configured using the **–link** option. The format of the **–link** option is as follows:

```
--link "ip=<remote  address> <tcp/udp>=<local port>:<port> watchdog=<timeout:retry> proto=<protocol>
```

Example(s):

- Host is behind a firewall that does not allow incoming connections. Create a link that will call out to another node or desktop outside the network when the node is launched.

  ```
  $ flx-packer -o node-linux -s linux:x86 -k aaaabbbb12345678 \
    -n kung-foo -m 1300 -i 1200:30:on --link ``ip=www.open.com tcp=8888 watchdog=2:4''
  ```

Combine both SP and Links depending on the situation the node is being built for

14.2 Edit linux-fluxwire.conf

CHAPTER

FIFTEEN

CHANGELOG

15.1 Release 3.5.0 (November 13, 2015)

15.1.1 Backend

- Fixed Mikrotik TileGX memory alignment issues specific to protocol filters.
- Added Freebsd 7.x and above platforms for 64bit architecture. Previous versions only included support for Freebsd 9.x and above for 64bit architecture.
- Added linux armv5-le support including embedded DVR platforms.
- Modified embedded library name of DLL to default to library.dll.
- Bundled libutil.a into shell module to support mixed versions on embedded armv5 platforms.
- Added work-around for broken uClibc versions that fail to spawn pthreads when daemonizing.
- Removed the full path of the PDB generated by the Visual Studio compiler.
- Obfuscated the embedded self-delete batch file for Windows executable nodes.
- Removed environment override of packer configurations for Device module. Operators must use the packer to configure nodes and will not be allowed to adjusted configurations on the fly using system environment variables.
- Fixed buffer checks on message filter in the Arsenic chain.
- Added HMAC to validate and authenticate all link headers. The header contains the CRC32 checksums for link payloads protecting against invalid packets moving further up the stack.
- Removed unnecessary link connect authentication due to new HMAC validation.
- Fixed immediate cleanup of new stack event inputs that fail due to invalid packets.
- Added independent watchdog for node shutdowns. The watchdog will monitor for shutdown in-progress and trigger immediate process exit for nodes unable to complete the shutdown in sufficient time. The watchdog component applies to executable nodes including node and archive configurations.
- Adjusted pipe names for Loader module to utilize the well known GUID identifier.
- Enabled priority support for memory loading all plugins on Linux x86/x64 and Darwin x86/x64 nodes. Nodes for these platforms will load all plugins in memory and never leave shared objects on disk even in an unlinked state.
- Added checks to RTP filter header fields and sizes during decode of the protocol packets.

- Refactored execution of main/core components and signal handlers for both node and desktop. Signal handlers behave inconsistent on Solaris and Darwin platforms requiring a generic implementation to address platform specific quirks.
- Fixed Transport module zero memory allocation bug for bridge and channel syncs. AIX does not handle zero sized memory allocs consistent with posix standards.
- Added additional DNS filter and Base64 checks to prevent invalid packets.
- Added additional SSL filter size checks during decode of buffers.
- Added HTTP filter checks preventing invalid packets.
- Fixed stack event reconnects spawning over and over on failed events/channels.

15.1.2 Control-API

- Fixed callbacks to wait for device stack address and added new example to show callback usage with disk transfer.

15.1.3 Packer

- Fixed the ‘blacklist’ option to properly parse and verify plugins passed in on the command line.
- Added exception if we can’t find the signature inside of the node or can’t find a node with available space.
- Changed Mikrotik 5.x/6.x platform compatible nodes to be explicitly defined by packer options.

15.1.4 Installer

- Updated installer to adjust permissions on all installed files. This change now allows users without root privileges to leverage flx-packer.

15.2 Release 3.4.1 (July 22, 2015)

15.2.1 Backend

- Fixed in HTTP protocol when handling authentication errors. This issue impacted http proxy links.
- Fixed bug in 3.4.0 that broke Device Alarm’s shutdown timer. The delimiter change in 3.4.0 caused a parsing error between backend and flx-packer.

15.3 Release 3.4.0 (June 19, 2015)

15.3.1 Backend

- Replaced all internal interfaces between modules and utilities with new plugin manager and new dynamic dependency loading. This change impacts ALL components, utilities, and modules on the backend.
- Fixed bug where offline AIX, OS X, and Solaris nodes were not being included in client interface’s initial offline node synchronization when an API or Horizon client connect.

- Fixed a locking issue during client connection that caused rare deadlock in flx-desktop. The issue was reproduced when many nodes are coming online all at once while an API or Horizon client is connecting.
- Added support for IPv6 services and links to all platforms that support IPv6 networking. Support for single IPv4 or IPv6 and dual-stack IPv4 and IPv6 services where added to platforms capable of supporting the modes. Some platforms do not support IPv6 at all, other platforms do not support IPv6-only single-stack, and some platforms may not support dual-stack.
- Modified Transport module to work with changes to data structures that were modified for IPv6 support. Transport module still *only* supports IPv4. Usability of Transport module should appear unchanged.
- Fixed some invalid memory copies in Transport module.
- Fixed bug where dash (-) was not be accepted as part of a valid remote hostname for a packed in link on a node.

15.3.2 Packer

- Refactored all of the source code base to simply several areas.
- The delimiting character was changed from colon to vertical pipe in the packer configuration string.
- Added support for IPv6 to services and links. Both have a new optional argument, ‘IP Version (ipv=)’, which defaults to a value ‘4’ (IP version 4) for both services and links. For services, ‘ipv’ can be set to ‘6’ (IP version 6) or ‘dual’. For links, the remote host ‘ip4=’ argument was replaced with ‘ip=’. The IP version is specified with ‘ipv’. For links, values of ipv include ‘4’ (IP version 4), ‘6’ (IP version 6), ‘6,4’ and ‘4,6’. ‘ipv=6’ will only resolve domain names using IP version 6, and ‘ipv=4’ will only attempt domain name resolution using IP version 4. ‘ipv=6,4’ requests that a successful IP version 6 domain name resolution be chosen over an IP version 4 resolution. Likewise, ‘ipv=4,6’ choose IP version 4 over IP version 6.

15.3.3 User Interface

- Added support for IPv6 services and links. Services and Links both default to IPv4. The Service and Link Panels now includes a combination box for IP Version.
- Added support for IPv6 link-local addressing by using network interface to indicate the scope id, e.g.: fe80::ca2a:14ff:fe24:24f9%eth0
- Fixed a bug that could balloon memory when Horizon’s Teardown Network feature is used. The bug happened when there are many loops and possible paths for nodes to reach the flx-desktop.
- Changed Visual Map screenshot feature to not refresh before taking the image. The refresh would cause nodes to move. Now the image matches exactly what the user sees in the Visual Map.
- Changed default font to DejaVu Sans Mono and changed Service and Link Panel tables to be monospaced to make IPv6 addresses easier to read.
- Fixed a bug where pool and pool timeout fields in LinkPanel would reenable while UDP link type is selected.
- Pool and pool timeout properties were added to existing links and are displayed in the Link Panel’s table.

15.3.4 Control API

- Added IPv6 support to Control API. A new named parameter, ‘ipv’, exists in create_service and create_link functions for device module. Both functions accept a default value of ‘4’, which indicates use IP version 4. See packer’s release notes for the values for the ‘ipv’ argument.
- Added pool_size and pool_timeout properties to link items to show the pool and pool timeout values configured on each link.

15.4 Release 3.3.1 (May 7, 2015)

15.4.1 Installer

- Java: Updated to Oracle Java Runtime Environment version 1.7.80-x64 to address language pack issues on Debian and Ubuntu platforms. Language packs must still be installed on each platform.

15.4.2 Backend

- SSL: Fixed segmentation fault if an alert protocol message was issued by the SSL client. Changed the format of the server certificate to use DER when transmitting over the wire. Fixed server hello message to include NULL compression byte and added additional checks for SSL type when processing data or handshake messages.

15.4.3 Packer

- Fixed an issue when passing full paths to the output option for ICE compatible module nodes. ICE loadable modules require a corresponding DLL name in the META.xml file and the resulting full path was not stripped into its basename component. Packer now strips the output path leaving only the basename in the META.xml file.

15.5 Release 3.3.0 (February 25, 2015)

15.5.1 Installer

- .bsx files have been replaced in favor of a more flexible .tgz package. The installer is now a python file (install.py) which is executed to install Fluxwire on the current system. Multiple versions on a single box are now supported and can be managed through ‘flx-manager’. The ‘flx-manager’ application allows users to switch between installed versions on the current system, uninstall old versions, list installed versions, and export a fresh .conf file for the current default version.

15.5.2 Backend

- Platforms: Converted Microsoft Windows builds to using CMake. Microsoft Windows nodes and plugins are now compiled natively using Visual Studio 2012. Using the Microsoft supported compiler helped decrease overall binary sizes for both nodes and plugins.
- Platforms: Added Linux Mikrotik node using Tile-GX architecture (experimental). Mainboard and netiface plugins have not been ported to this platform.
- Disk: Fixed UID/GID values being potentially incorrect on Darwin. Most systems define UID/GID to be a uint16_t but Darwin defines them as a int32_t. On Darwin the user/group ’nobody’ is set as value -2 (unsigned value of 4294967294) and we were not able to accommodate this value prior to this fix.
- SSL: Added support for using straight SSL as a protocol filter option. The SSL filter protocol can be used in full encryption mode or a lightweight mode. The lightweight mode will only perform the steps for the SSL handshake and the payload data will not be encrypted. The full encryption mode is the same as the lightweight mode, with the exception that, it will encrypt the payload data with the cipher chosen in the handshake.
- SSL: Added support for using the PolarSSL and OpenSSL libraries. The PolarSSL library (v1.3.7) is bundled into the PolarSSL version of the SSL plugin. The OpenSSL version of the SSL plugin operates the same as in previous releases.

- Loader: Added ability to leave loaded modules running for a specified interval of time if the node goes offline. C2 will re-sync and retransmit any data messages that did not get received when node comes back online. A disconnect timeout value (minutes) can be specified and controls how long both sides wait before closing C2 and unloading loaded modules. New loader states: C2 Down, Resuming, and Exited while offline.
- Core: Added a pool of TCP connections per link where a set of connections are rotated through and made active for a set period of time. Once a connection timeout has been reached the next connection is activated and this process is repeated. The maximum number of connections kept at one time can be specified through the 'pool size' option. The 'pool timeout' option is used to set the minimum duration of a connection before its replaced with another connection. This feature does not support UDP. HTTP(s) uses a pool of underlying TCP requests as it did previously.
- Device: Fixed identity name to better support Unicode characters by converting the storage identifier from string to bytes.
- HTTP(s): Changed client user agent string to match customer requested user agent.
- All: Changed time calculations to use 64-bit throughout Core and all plugins to avoid 32-bit rollover on certain platforms.
- Loader: Fixed a customer reported bug where a race condition existed between Control API creating new loader instances before a target node has finished synchronizing with Controls during plugin load.
- Loader: Changed command response acknowledgements to always come after new listing messages that alert Control API and Horizon to new instances or state changes in loaded instances. This simplified Control API's responses.

15.5.3 Packer

- SSL: Added PolarSSL support for POSIX systems and SSL filter options for full and light modes.
- SSL: Fixed SSL link only configurations not including the SSL filter plugin.
- SSL: Added support for choosing between PolarSSL and OpenSSL on Posix systems.
- SSL: Added validation checks on SSL and HTTP(S) packer options.
- Device: Added validation check to device shutdown alarm to make sure timeout value at least twice that of the interval.
- Device: Added configuration support for links with a pool TCP of connections.
- Loader: Added support for Windows 8.1 to the ICE specification XML file.
- Install: Changed default install path to /usr/local/fluxwire/default.

15.5.4 User Interface

- SSL: Added PolarSSL support for POSIX systems and SSL filter options for full and light modes.
- Disk: Fixed UID/GID value size to support 32-bit values due to how Darwin stores these values.
- Transport: Added a default gateway badge to the Network Window.
- Device: Fixed bug when changing a device's Identity Name when using unicode characters. The number of characters were being counted instead of the number of bytes (max 64) causing the potential for characters to be displayed incorrectly on a name change. If more than 64 bytes are entered the user is warned of this and cannot change the name of the device until this issue is resolved.

- Device: Support for links with a pool of connections has been added which includes the following changes: ‘pool size’ config option has been moved to the Link Panel, ‘pool size’ forced to 1 for UDP, 3+ for HTTP/HTTPS, and 1-8 for all other TCP connections, added pool timeout spinner to Link Panel, and removed ‘pool size’ spinner from HTTP(s) link options dialog.
- Loader: Added support for the new feature to leave loaded modules running for a specified interval of time if the node goes offline. New spinner added to the Call Dialog for disconnect timeout (0 is off).
- Transport: Fixed a bug when adding transport rules via Horizon using “x.x.x.x/yy” notation would be interpreted as “x.x.x.x/32” in the backend.
- Disk: Fixed bug where Touch dialog was accessible on Microsoft Windows nodes when multiple files were selected.
- Disk: Fixed bug with Go To Directory Dialog where dismissing the dialog without accepting the path but still caused the path to be accepted and requested.

15.5.5 Control API

- SSL: Added PolarSSL support for POSIX systems and SSL filter options for full and light modes.
- Device: Added new network teardown function to device interface.
- Device: Added graph support. Usage requires networkx and pygraphviz python libraries and graphviz. Supports graphviz node and edge options via dictionary or function and supports all available graphviz formats. Added Transport SessionModelItem, which can be viewed on the graph via session filter.
- Documentation: Updated documentation and added new examples.
- Device: Added support for links with a pool of connections: pool_size and pool_timeout have been added to create_link().
- Loader: Added support for the new feature to leave loaded modules running for a specified interval of time if the nodes goes offline. There is a new parameter, disconnect_timeout, in loader validate function that takes a timeout in minutes.
- Transport: Added a function to return the default gateway for the Transport module.

15.6 Release 3.2.1 (August 18, 2014)

15.6.1 Backend

- Fixed system descriptor ids that were out of sync. This issue caused MIPSLE, MISPBE, ARMLE, and ARMBE processor architectures to report incorrect identities in Horizon Identity Panel.
- Fixed bug where an allocated socket would not be freed if multiple socket connections failed rapidly.
- Fixed HTTP proxy basic authentication support.

15.6.2 Packer

- Fixed HTTP proxy basic authentication configuration support.

15.6.3 User Interface

- Added label for reporting "ARMBE" architecture and "Backfire" model in Configure Window's Identity Panel.

15.7 Release 3.2.0 (July 15, 2014)

15.7.1 Backend

- HTTP(S): Added support for transparent and manual proxies. Configuration options were updated and simplified for both service ports and links. Service ports allows selection from predefined HTTP protocol server response headers. Link options now include a flag to enable proxy. HTTP host can optionally include a host port to form HTTP requests targeting non standard HTTP port e.g. www.ixmb.org:8080.
- Transport: Fixed bridge updates not properly being duplicated to more than the first client session.
- Platforms: Fixed bug reported by customers concerning unresolvable libgcc library dependency on SUSE Linux 8. The fix was applied to all Linux and OS X platforms.
- Core: Changed flx-desktop to allow coredump files to be generated to help with future bug reports.
- Core: Fixed race condition where the control-api attempts to connect to a Control before it has completed its setup and causes a crash.
- SSL: Fixed MingW compiler typecast issue for big endian conversion of encrypted buffer impacting MSWIN to MSWIN devices.
- Packer: Added proxy support with updated HTTP configuration options.
- Packer: Adding missing configuration items to HTTPS.
- Packer: Updated DNS protocol so that domain can be omitted.
- Fixed issue reported for airOS platforms being unresponsive and links constantly failing or timing out. When message buffers would fill completely up the data would quit flushing causing timeouts.

15.7.2 User Interface

- HTTP(S): Added support for proxies. Service and link options dialogs for HTTP and HTTPS were changed for new configuration parameters.
- Transport: Fixed bridges where socket bind errors reported for duplicate create bridge requests falsely associated with the active successfully created bridge.
- Device: Added module operations to Visual Map right-click menu under a "Network" submenu label. When Device Configure option is selected in Visual Map it has to force selection in Network Window.
- Device: Changed offline Visual Map image generation to cache and reuse images for 10 seconds to lower memory consumption. Also, images are now discarded explicitly if an offline device comes back online.
- Device: Fixed collapsed links on Visual Map lingered after all links were removed between two devices. This issue presented when multiple links were created between two devices and then subsequently removed.

15.7.3 Control API

- HTTP: Added proxy support with updated configuration options.
- Device: Added pool_size option to create_link.

- RTP: Fixed the configuration options so the key-value format is used.
- DNS: Updated the configuration options so the domain will be omitted if not explicitly set.
- Device: Fixed an issue where an incorrect link or service could be returned from the create functions.
- Transport: Fixed the create_rule function so that it can find link types, accept wildcard values, and allow for ranges and comma separated values.

15.7.4 Packer

- DNS: Fixed DNS protocol to not include keys that have not been set in the configuration string (i.e., domain= was being included even when it had not been set).
- HTTP: Added proxy support and new HTTP configuration parameters. Added pool option to device link configuration.
- HTTPS: Added missing config item (header) to HTTPS.

15.8 Release 3.1.1 (May 14, 2014)

15.8.1 Backend

- Device: Fixed nodes startup failure reported on MipsBE platform, though bug was possible on all platforms. The issue was that getting the hostname at startup could cause a thread to block too long for a network ARP request which caused the monitor thread to timeout and abort the Device module.
- Core: Fixed control-api timing issue where connections prior to control completing its setup would result in a crash.

15.9 Release 3.1.0 (February 20, 2014)

15.9.1 Backend

- Loader: Implemented a new Loader module to support version 3 of ICE specification's new featureset: 1) Added staging cache on Controls for DLLs loaded from Horizon or python Control API. 2) Add support for C2 support for Collect or Interact behaviors. 3) Added support for processing unload deployed module (DLL) requests.
- Loader: Added Loader support to python Control API.
- Loader: Added debugging feature to log C2 message passing in PCAP format.
- Core: Added SHA256 hash support for loader to support ICE version 3 metadata files.
- DNS: Fixed a memory leak.
- Platform: Added Mikrotik 6.x for mipsbe, mipsle, ppc, and x86 architectures.
- Platform: Added FreeBSD 9 on x64.
- Desktop: Changed desktop from Linux x86 to Linux x64.
- Core and Transport: Fixed a link timeout bug that could have caused links to linger and possibly Transport routes automatically created by Fluxwire to not be removed.
- Device: Fixed a bug where local peer address ports were not being set or byte swapped properly that caused wrong values to be displayed in Horzion.

- Device: Added the ability to set the device name to hostname on startup when the device name is not provided.
- Disk: Fixed NULL pointer segfault and added a new check to ensure file sizes don't exceed the maximum size limit allowed.

15.9.2 User Interface

- Horizon: Added support for Ubuntu 12.04. Fixed textfield sizes so underscores do not blend with the bottom of field in Ubuntu 12.04.
- Horizon: Updated some dependency libraries to newer versions.
- Loader: Added new implementation for Loader module to support ICE specification version 3: 1) Created a staged table to show DLLs in staging cache. 2) Implemented a new Call Dialog to support Collect or Interact behaviors C2 handling. 3) Added ability request loaded DLLs to unload.
- Device: Added hostname to the Identity Panel.

15.10 Release 3.0.0 (August 23, 2013)

15.10.1 Backend

- Chat: Fixed customer reported issue of chat potentially causing the desktop to crash in certain situations.
- Control API: Added the Python API for Controls with support for for asynchronous or synchronous messages, exception handling, and support for device, disk, shell, and transport modules.
- Core: Decoupled event and link layers in the Fluxwire stack using a newly developed Arsenic IO library.
- Core: Added performance enhancements to speed up message handling.
- Core: Added the following Arsenic filters: 1) MSG filter - Wraps all data being sent in a header. This filter is only used in TCP connections and allows us to support stream oriented data. 2) Socket filter - Handles all reads/writes from/to a socket
- Device: Fixed buffer overflow and contamination of link protocol configuration when watchdog is set.
- Filters: Ported DNS, HTTP, and RTP to an Arsenic IO filter which will handle encoding/decoding the data according to the protocol specification.
- HTTPS: Removed HTTPS and replaced it by a filter chain consisting of HTTP and SSL filters.
- Packer: Ported the C version of packer to a Python library and command line client. The result is a new binary that provides a simpler and cleaner interface to configure and build nodes.
- Platform: Linux Mipsbe AR71xx, Linux Mipsbe Airosv, Linux Armle GP, Linux Mipsle CPE, Linux Mipsle Inca, and Linux Mipsbe Inca.
- SSL: Arsenic filter that will handle encoding/decoding SSL data.

15.10.2 User Interface

- All: Converted unicode characters to their equivalent encoding to fix an issue with certain characters not rendering properly.
- Device: Added the missing model IDs (ar71xx, airosv, gp, ...).

- Device: Fixed the crytpo library path for HTTPS service port options to be editable and not use a file select dialog.
- Disk: Added the Go To Folder dialog that allows a user to enter a valid system path and jump to the given directory, if it exists.

15.11 Release 2.4.1 (February 28, 2013)

15.11.1 Backend

- Platform: Added a fix to solaris sparc to remove debug symbols.
- Platform: Fixed the strip commands on some makefiles to remove unneeded sections.
- Platform: Added support for linux/mipsbe/airosv.
- Platform: Added support for linux/mipsbe/ar71xx.
- Packer: Fixed an issue where multiple spaces between config options was causing packer to fail.
- Disk: Fixed Linux length calculation of mkstemp() file name string.
- HTTPS: Fixed crypto search paths for 32bit node on x64 system.
- HTTP: Fixed an infinite loop issue when http->length was set, but the amount of data in the buffer was smaller then http-length it would return 0 and size in link_http_input would never get updated.
- Packer: Fixed an issue in packer where it would continue packing even if a plugins dependency was not found.
- Stack: Fixed route sync ordering of bundles to prevent unreachable entries.
- Loader: Added the ability to import functions by ordinal
- Transport: Fixed large memory leak in transport database on Controls.
- Scripts: Updated version in .conf files and added missing sections into default.
- HTTPS: Added HTTPS as a filter.
- Shell: Updated busybox binaries on mswin and fixed a small memory leak on plugin.
- DNS: Fixed memory leaks during a failure case.
- Disk: Fixed race condition between transfer database thread and overlay thread.
- Log: Added file_close call for non-stdout handles and memory_free on file handle.
- Device: Fixed memory leaks on shutdown.
- Core: Fixed minor memory leaks in core.
- Disk: Added hash to db_html script.
- Chat: Fixed session memory leaks.
- Disk: Fixed crash bug when reloading Disk plugin on local controls.
- Device: Added new link watchdog feature.
- Disk: Fixed issue where a walk request does not fail and remove when a link goes down.
- Stack: Fixed link optimal algorithm to include incoming ack time.
- Platform: Added support for linux/mipsle/cpe.
- Platform: Added support for linux/mipsle/inca.

- Platform: Added support for linux/mipsbe/inca.
- Platform: Added support for AIX PPC and PPC64 nodes.
- Disk: Added the “auto_refresh” configuration option.
- Disk: Added the hash file operation with MD5 support.
- Platform: Fixed gcc 4.7+ packing of data structure issues.
- Fixed: Peer session was missing for indirect peers when the route entry was allocated.
- Stack: Added mesh session handler for initializing peers.
- Stack: Added updater to finding optimal link for mesh routes.
- Stack: Fixed out-of-band processing of commands and hard timeout for link connects.

15.11.2 User Interface

- Shell: Changed single command text area to escape commands on Windows devices.
- Chat: Fixed null pointer exception in Chat Invite Dialog.
- Dashboard: Moved the Network Shutdown button to appear next to the Power button.
- Device: Disabled multiple selection of plugins in plugin panel.
- Device: Added https filter support.
- Shell: Added Text Area to allow single command line to be sent in one message to the terminal.
- Disk: Updated log4j file.
- Horizon: Changed Log Window sorting to put most recent messages at top of window.
- Horizon: Fixed uptime timer deadlock in dashboard.
- Device: Fixed exception in Plugin Panel when device list or platform descriptions load improperly.
- Device: Added Watchdog configuration components to Link Panel.
- Horizon: Updated how the uptime is calculated in the Dashboard.
- All: Added support for AIX nodes - icons and system info
- Disk: Modified the directory window to reject online directory messages in offline mode, and vice versa.
- Disk: Automatically refresh the directory window when all file operations are completed.
- Disk: Added support for the hash file operation.

15.12 Release 2.4.0 (September 25, 2012)

15.12.1 Backend

- Platform: Added MAC OS X x86 and x64 nodes with supporting uid plugins.
- Platform: Refactored code to optimize node binaries for size across all platforms.
- Transport: Added periodic check to find deleted user routes including the “full routing” mode default route.
- Desktop: Updated desktop startup notifications.
- Core: Fixed plugin monitor for threads failing to return in a timely manner.

- Disk: Added the ability to modify file timestamps within scope of the privileges.
- Disk: Added support for multiple clients connected to a single control.
- Disk: Added the ability to handle/process directory transfer requests
- Node: Added cross platform path detection library using multiple techniques for determining execution path.
- Disk: Fixed memory leak during plugin load and unload notifications on Controls.
- Disk: Fixed memory leak in Disk database storage.
- Core: Fixed a memory leak that would occur if a link could not be established
- Node: Added self-delete counter and limit for MSWIN nodes.
- Disk: Fixed memory bug and leak for handling link targets in directory requests and file information respectively.
- Node: Moved posix self_delete to handle edge case embedded devices where main thread disappears.
- Transport: Fixed clearing udp buffer for packets match deny policy rules.
- Loader: Fixed command line arguments parsing using maximum set length.
- Disk: Added link target path resolution of symbolic links, junctions, and .lnk files
- Shell: Added /sbin/sh path to shell execl list for Solaris 10.
- Device: Added node granularity blacklist option for blocking specific modules and plugins.
- Device: Added priority flags to all packets for new priorty channel.
- Core: Added priority channel for link and mesh layer packets.
- Core: Fixed release of packet for receipts only when references is zero.
- Disk: Fixed an issue in where tz_minuteswest was not converted to seconds for Get Info requests.
- DNS: Added DNS protocol to the mesh netwoking stack.
- Device: Changed device database to use prepared statements and made several fixes.
- Core: Optimized plugin loading protocol to trim packets for reply messages.
- SQLite: Updated to version 3.7.13
- Disk: Removed fullpath from filesystem table in disk.db to improve performance/space and added SQLite profile/trace calls.
- Netiface: Added FreeBSD support for MAC interface.
- Mainboard: Added FreeBSD support for inspecting hardware.
- Disk: Fixed memory overflow error when saving m/a/ctime in db_get_file_info()
- Disk: Added ‘file get info’ optimization feature
- Node: Added rundll32 support for Windows platforms.

15.12.2 User Interface

- Transport: Fixed issue with route filter not showing up in Routes Panel after leaving route conflict resolution mode.
- Device: Changed Plugin Panel to deny unloading Disk and Transport plugins on Controls.
- Device: Fixed Link Panel so failed link connections do not show up when browsing offline devices.
- Horizon: Added new global title bar icon and fixed font type/size issue in window titles

- Device: Changed titles and dialog sizes for protocol options to fit with bigger fonts on Linux.
- Device: Removed Network submenu from Visual Map.
- Device: Added support for DNS protocol to collapsed edge labels in Visual Map.
- Disk: Modified the Download menu item so it will be disabled for all types of links and fixed missing character in mode of empty directories
- Transport: Updated the length of the node's name in the status panel to ensure that it does not run off screen.
- Disk: Fixed bug with permissions button being disabled in Directory Window.
- Disk: Fixed bug disposing of table model components that caused an exception in Directory Window.
- Horizon: Added min width to certain Text Fields to prevent them from going offscreen when filled with more data than can be viewed.
- Horizon: Fixed DocumentLimit so replaced text is also validated.
- Disk: Fixed UI controls enabling and disabling in Directory Window.
- Device: Updated diagnostic panel to display the node's name if its available instead of the ID.
- Loader: Added unique name to argument text area and set a limit of (260 * 4) - 1 characters
- Device: Added blacklist error message for plugins.
- Disk: Added the ability to follow links in a Directory Window
- Transport: Added Horizon modal dialog to warn when full routing is disabled by external application.
- DNS: Added the DNS Protocol Options Dialog for UDP links in the Link Panel.
- Transport: Added route type filtering to Route Panel Preference Panel.
- Transport: Fixed bug where device identity label was not updating in Transport Status Panel when identity changed in Device Configuration Window.
- Device: Fixed HTTP link options dialog was accepting an empty host string.
- Device: Fixed Visual Map to default to Local Control as Radial Map center when there is no device selected.
- Device: Updated the tables in Link, Plugin, and Service panels to match the l&f in the rest of the GUI.
- Horizon: Added offline node listing, offline device information, offline disk browsing, and drag and drop capabilities to Network Window.
- Disk: Changed data storage to reduce memory consumption in Directory Window.
- Device: Changed Visual Map screen shot's Save Dialog so directories appear in its listing.
- Disk: Fixed bug where database saving icon sometimes appears after completed icon.
- Device: Changed HTTP options dialogs to use auto complete combo boxes.
- Transport: Added support for network/submask notation for network ranges in Add Rules Dialog.
- Disk: Added the file 'touch' user operator controls.
- Horizon: Added OS Icons in the frame of Disk Directory and Shell Session windows
- Disk: Added the ability to transfer directories using metadata stored locally about remote filesystem.
- Disk: Update protocol file, updated file 'get info' to work with new db change, and fixed order of calls when saving basic file info objects.
- Disk: Fixed implicit calls to dialog box disposal code when Directory Windows are closed.
- Disk: Fixed duplicate dispose call in Directory Window when closing it.

- Horizon: Added editable combo boxes with the ability to display a history of validated entries.
- Device: Added new features Radial Map to show minimum number of hops between Nodes.
- Device: Added Teardown Network feature to shutdown all nodes in the network.
- Device: Fixed disabled host textfield in Link panel when link is removed and updated Link/Service panel layouts.
- Horizon: Added new FocusedMenuList to fix MenuLists in global windows from appearing disabled.
- Horizon: Added DeviceDialog to make dialogs operating on a control/node appear uniform.
- Horizon: Fixed issue with long names in a title bar
- Substance: Updated substance .jar files and included javadoc and source for each .jar
- Disk: Fixed dispose null exception and added dispose() calls to the dialogs in a Directory Window.
- Horizon: Added Horizon.java class as the main entry point.
- All: Added new title icon to all frames/dialogs
- Disk: Added optimization for ‘file get info’ to only request info during user request.
- All: Added jxlayer glasspane for managing notifications and user input.
- All: Changed the JTextFieldValidator for better performance by requiring less GUI rendering.
- Device: Added ability to access Network Window’s menu from Visual Map.
- Device: Changed Network Window and Visual Map to display a public IP if one exists in the node’s set of links.
- Device: Added the ability to reorder Nodes and Controls in the Network Window by dragging and dropping.
- All: Fixed a bug in the Delete Dialog by clearing the JList when the dialog is hidden.
- All: Added custom JTextField search component and updated all classes that implement a custom search field.
- Device: Fixed misspelling in HTTP Link Options dialog.

15.13 Release 2.3.0 (May 23, 2012)

15.13.1 Backend

- Mainboard: Modified MSWIN architecture to use registry calls to avoid popups by PSPs. Using the native WMI interface and SetupDi functions were calling AdjustTokenPrivileges to try to set SeDebugPrivilege, all of which were triggering Kaspersky.
- Packer: Added a check to ensure the entire device config section is processed during command line node building.
- Core: Enhanced desktop and node with higher file descriptor limits to avoid exhaustion.
- Disk: Added temporary file cleanup logic when “open” fails during a new download job.
- All: Added client session signal to support disconnect in addition to a connect from a client. Internal change helps the backend organize metadata associated with user interface clients.
- Disk: Fixed logic associated with queue pending counts for file transfers to support repeated link connect and disconnects to the same node.
- Disk: Added file handle cleanup for file transfers not currently active. Handles are requested just in time for the transfer.

- Disk: Added the continue flag to directory read and walks indicating to user interface clients the end of the directory has been reached.
- Disk: Added file name increment for downloads if a local control has a name collision.
- Transport: Added support for client configure option for UDP timeout on outbound connections.
- Core: Fixed the reference count associated with datagram events. Datagram events spawned from an outbound connection combined with an inbound connection against the same local port resulted in an incorrect usage counter. If the first link went away then the associated event is zeroed out.
- Device: Added interface function for all other modules to retrieve UID hash using the stack ID as a parameter.
- Device: Added human readable hash (MD5) of the variable length UID block to the sqlite table.
- Sqlite: Added the bind_int64 function to the sqlite interface.
- Disk: Added support for tracking filesystems in the database with hardware identifiers instead of stack identifiers.
- Device: Added locale and timezone information and interfaces. Device provides locale data (when available) and timezone data minus the daylight savings time.
- Disk: Updated file time conversion to use the proper timezone retrieved by the device. Access, modified and creation times are now adjusted automatically to the device specific timezone data.
- RTP: Added real-time transport protocol to the stack filters. RTP supports two encoding schemes: PCMU and PCMA.
- Node: Added DLL loadable support supporting the MRC fire-and-forget interface, rundll32 and init with no args.
- Transport: Added check and set boundary conditions for UDP timeouts. Both config file and user interface updates to UDP timeout are checked against lower and upper boundary conditions currently set to 2000ms and 3600000ms.
- Device: Fixed device database for links to lookup protocol using type as a dependent. Protocol string values were looked up independent of type resulting shared values for HTTP and RTP.
- Chat: Added basic and group messaging with notifications to support secure operator inter-network communications.
- All: Added cleanup of in-memory copies of unique strings to prevent crash-dump analysis.
- Transport: Fixed memory leak when user or system route list client request message is received by controls.
- Transport: Added rule feature for customizing traffic routing policies for new connections using IP addresses, ports, and link type (TCP or UDP). Operators can designate specific nodes for certain types of traffic based on the rule sets.
- Device: Fixed link property lookup to match only id and remote address. The link property locate function was matching against id, external and remote addresses. The external address does not get updated until after the connection is at CONNECTED state resulting in a quick link remove and link add during state changes.
- Desktop: Added startup notifications and fixed daemonize to allow standard output.
- Loader: Added generic DLL loading capability in support of the MRC fire-and-forget interface defined by ordinal type 1. Loader supports in-memory loading of the DLL, validation of PE structure before loading, and progress messages during load, activation and call processes.
- Transport: Fixed bridge bind issue on MSWIN platforms. Bridges were setup without SO_REUSEADDR (on MSWIN only) and if a connection is accepted, the new connection had SO_REUSEADDR set. At this point if the old bridge is disabled and restarted, the bridge will error out with ADDRINUSE.

- Transport: Fixed bridge listener socket options ordering. Socket options for non-blocking and reuseaddr need to come before the bind according to Microsoft Windows platform winsock specification.
- Core: Added SetErrorMode(SEM_FAILCRITICALERRORS) to prevent error mode dialogs from hanging the node on Windows. Setting this will prevent the system from displaying the critical-error-handler message box. Instead, the system sends the error to the calling process.
- Disk: Fixed a logic bug when using 'LIKE' in a database statement to mark files.
- Transport: Fixed link enabled routes for loopback, broadcast and certain masks. Link enabled routes for not properly excluding the loopback range and broadcast address. Default configuration needs to utilize a /32 mask for single IP networks.
- Packer: Updated help examples for building a node.
- Platform: Added experimental FreeBSD 7.x-9.x support for nodes.
- Disk: Fixed offline browsing support for when the hardware unique id is unavailable. Disk database prefers the hardware uid hash, but fall backs to the stack address.
- Disk: Updated the db_split.py script to handle both UID and Stack ID values in the devicecache table. If the UID value is not found then the Stack ID will be used to identify the DB.
- Disk: Fixed local control queue logic for decrementing the count. The local control active queue count was being decremented by file transfers started by the local control as well as file transfers that did not belong to the control. Added check to ensure that only file transfers that were started by the local control affected the count.

15.13.2 User Interface

- All: Added user input validators to text fields for Dashboard, Device, Disk and Transport modules. Validation of all input against regular expression rules to ensure operators provide correct values.
- Disk: Added a check to determine when to disable the progress bar on the directory read and walk.
- Disk: Changed double clicking in the Directory Window to be only available for left mouse button only.
- All: Updated the renderers to ensure the anti-alias option was enabled for smooth font rendering.
- Device: Added timezone and locale display fields in the Identity Panel.
- Device: Added RTP configuration options for service ports and links.
- Transport: Added preference panel to Transport Window to configure the UDP timeout on outbound connections.
- Device: Added screenshot feature for network diagrams on the visual map.
- Chat: Added user interface in support of one-to-one and group messaging, scrollback buffers, and notifications for invite, typing and exit.
- Transport: Added rule management interfaces for customizing traffic routing policies on all new connections. Added support for rule import and export using space delimited text file format, drag and drop reordering, multi-value support using comma delimited as well as ranges, dialog for creating new rules, and custom renderer to highlight default rule.
- Device: Updated the Link Panel refresh to properly reflect immediate changes from the backend.
- Loader: Added user interface for loading Microsoft Window DLLS on remote nodes. Supports load, unload, and call with real-time progress messages and notifications.
- Device: Updated Plugin Panel to filter out control plugins from being displayed on node plugin lists.

- Disk: Fixed logic error on enabling the progress bar when switching states (online/offline) in a Directory Window. If the Directory Window model is empty the progress bar should not be enabled when a user changes state.
- Device: Updated protocol/type names that should be in all capital letters (TCP, UDP, HTTP, RTP).

15.14 Release 2.2.3 (January 17, 2012)

15.14.1 Backend

- Disk: Added the '%P' option in the download path substitution feature to allow users to select the drive letter for Windows nodes.
- Disk: Added tmp file cleanup code if open fails during the creation of a new download.
- Core: Changed the max file descriptors for Darwin to 10000 for both desktop and node.
- Core: Added get/setrlimit() calls to desktop and added a call to getrlimit in node/main.c.
- Packer: Added a check to ensure the entire device config section is processed.
- Mainboard: Fixed an issue in mswin mainboard that was triggering a pop up by a PSP
- Disk: Changed number of files sent to GUI in one packet in offline browsing mode to increase GUI responsiveness.
- Core: Improved packet loss calculations for the stack.
- All: Fixed buffer and packet memory leaks.
- Shell: Added support for mswin style pathnames
- Core: Added TCP retransmit for raw links subject to EWOULDBLOCK.
- Disk: Fixed fwrite() memory use on Posix systems by assigning buffer size to the size parameter and the number of items to 1.
- HTTP: Updated the URI extension and depth and added browser signatures.
- Disk: Fixed token range bug where the token range was not being correctly calculated at certain times.
- Disk: Added a file transfer table to the disk.db to track file transfer requests and stats.
- Disk: Fixed memory leaks from some message handling on overlay and client interfaces.
- Core: Fixed an issue in the thread library for mswin builds
- Disk: Added the ability to query drives and drive type on Microsoft Windows nodes.
- Disk: Changed all file operation request handling to remove unnecessary memory allocation and release.
- Device: Changed error checking on incorrect pointer after allocation.
- Device: Added ability on mswin nodes to load as a dynamic link library.
- Core: Added local peer name capture
- Disk: Added the ability to pause/resume active file transfers.
- Disk: Fixed file transfer packet receipt bug.
- Disk: Fixed incorrect length in windows path char conversion when parsing out the %p option in the file transfer path.
- Disk: Fixed missing cleanup call in node to global registry if control goes down.

- Transport: Added new feature to scan and send system routes to Horizon.
- Disk: Added a loop to save off waiting download data packets if the source of the packets goes down.
- Disk: Fixed missing active file transfer increment when the max active value was set to 0.
- Device: Fixed missing pointer adjustment when processing device DB UID value.
- Scripts: Added missing .conf entry for maximum number of active file transfers in Disk.
- Disk: Added a file transfer control queue to provide more control over the number of active file transfers in the mesh network.
- Device: Fixed potential missing NULL in the device name buffer.
- Device: Added a device database to store information about devices, services, links, and hardware information.
- SQLite: Upgraded from version 3.6.17 to 3.7.8
- Disk: Added the ability to specify a maximum directory depth for directory walk operations.
- Core: Fixed minor problem with debug builds where file descriptor was not being checked if function map file could not be opened.
- Device: Added the ability to shutdown controls from the Horizon client interface.
- Shell: Changed format of shell session log files so that the date-time is first in finalized log file names.
- Device: Fixed host order conversion of the memory size in UID.
- Transport: Fixed issue where a duplicate default route could be added if control shuts down without ever enabling full routing feature, and fixed a bug where fluxwire's route table and the system route's table would get out of sync if an invalid gateway was specified in a user defined route.
- Disk: Added file transfer (download) path variable substitution support.
- Disk: Updated the string format of atime, mtime, and ctime to allow for more efficient sorting in Horizon.

15.14.2 User Interface

- Device: Removed the delete option in the Shutdown dialog for Windows DLL nodes.
- Device: Changed Visual Map's default state to show Control badge on nodes.
- Transport: Fixed minor user interface usability issues.
- Disk: Fixed several GUI issues when plugins are unloaded and reloaded.
- Device: Modified ordering of link information in Link Panel.
- Disk: Changed Directory Window so it clears listing contents when it closes to reduce memory consumption.
- Device: Added feature that highlights nodes with poor links with a pulsing yellow and red animation.
- Device: Added menu item to Device menu in Network Window to bring up Configuration Dialog.
- All: Updated all icons to be located in a central location (Globals.java) and updated references to these icons to improve memory performance.
- Device: Added new Visual Map features to customize information displayed and allow users to minimize clutter.
- HTTP: Increased URI max depth to be 25.
- Disk: Added the ability to query drives on Microsoft Windows nodes in Read and Walk dialogs.
- Device: Added new field to Identity Panel to display the binary type: Native exe, Windows DLL, etc.
- Device: Added display support for local peer name in Device Configure dialog's Link Panel.

- Disk: Added the ability to process and handle file transfer pause/resume actions.
- Disk: Fixed logic in the file transfer clear inactive button in the Transfer History panel.
- Transport: Added new feature to display system routes and resolve route conflicts for IPv4 routes.
- Disk: Fixed sort order in the Directory Window size column for files over 2 Gb.
- Disk: Changed directory walk max depth spinner to display zero by default for no limit instead of 8192.
- Disk: Added support for displaying and handling pending file transfers.
- Device: Added the ability to scroll through the device name JTextFields in the Device Configuration window.
- Disk: Added the ability to specify a maximum directory depth for directory walk operations.
- Device: Added ability to shutdown the Local Control and removed the shutdown command from remote Controls.
- Device: Added 64 character limit to the name field in the Identity Panel.
- Transport: Updated the Mask field in the Add Route Dialog to be a spinner instead of a combo box.
- Disk: Fixed issue where extra characters were showing up in the Permissions column of Windows nodes for certain versions of Java.
- Horizon: Fixed a minor bug where Log window was not being added to Dashboard's Window drop down box.
- Disk: Added node name replacement in the transfer download path when %N is encountered.
- Horizon: Fixed bug in uncaught exception handler that caused some valid exceptions to never present to the user.
- Device: Fixed an issue that caused new UDP link creation with the Link Panel to fail when a user specified local port number is supplied in the spin box.
- Horizon: Changed the default sort column of Log window to the alerts column and made alerts appear at the top of the window.
- Horizon: Fixed a race condition with clicking Connect button multiple times, and changed the connection failed logging message so that an exception stack trace is not included.
- Horizon: Fixed how combo boxes render device names to ensure that even if the device name is too long, it doesn't cause the combo box to be pushed off screen.
- Device: Added new system id to GUI for vanguard target.
- Shell: Added a 15 second timer that when necessary replaces the stack address with device name in the Shell Info table.
- Disk: Fixed issue where during a sync the wrong file transfer row was being updated due to multiple matches.
- Shell: Added support for new busybox version of Windows shell sessions and enabled shell session history for Windows.
- Device: Removed lingering println when creating a link.
- Horizon: Updated custom comboboxes to match the default L&F and size.
- Device: Added local port support for links in the Link Panel.
- Disk: Fixed Start Directory Walk button so that it displays an error message instead of the walk dialog if all currently connected nodes cannot be walked.
- Disk: Removed outdated code from the Directory Read Dialog.
- Transport: Fixed the height of the status bars in the global Transport window (route and rule panels).

15.15 Release 2.2.2 (November 28, 2011)

15.15.1 Backend

- Shell: Changed format of shell session log files so that the date-time is first in finalized log file names.
- Device: Fixed host order conversion of the memory size in UID.
- Node: Removed unnecessary printf() if the call to daemonize() fails.
- Transport: Fixed issue where a duplicate default route could be added if control shuts down without ever enabling full routing feature, and fixed a bug where fluxwire's route table and the system route's table would get out of sync if an invalid gateway was specified in a user defined route.
- Disk: Updated the string format of atime, mtime, and ctime to allow for more efficient sorting in Horizon.
- Core: Fixed a minor issue with clean up of a UDP link data structures.
- Shell: Updated MS Window's busybox executables to print command not found.
- Disk: Fixed improper usage of deprecated status code in overlay message handling.
- Shell: Fixed a potential memory leak when handling shell session operation commands in controls.
- Platform: Added Vanguard ARMLE support
- Http: Fixed link stream failing to send a disconnect acknowledgement.
- Core: Fixed link stream failing to send a disconnect acknowledgement.
- Disk: Fixed file transfer issue where remote Controls were not receiving status change messages.
- Shell: Fixed bug where shell logging feature was initializing properly during initial sync command.
- Shell: Converted busybox from a multi-process model to a threaded model
- Shell: Added device identity name to shell terminal log files.
- Device: Added device identity name to stack description structure so modules could use it for file names.
- Shell: Added shell session logging feature for each shell session's active control.
- Core: Added ability to run desktop and nodes as daemons on Posix systems.
- Packer: Fixed LK examples where the local port was missing.
- Packer: Added local port binding support to both Packer and Device.
- Packer: Fixed segfault when http option with no config name and one or more http configs were defined.
- Scripts: Added two new disk database parsing scripts into the scripts/tools directory.
- Disk: Fixed a build error related to a lingering #ifdef DEBUG that was not needed in xfer_upload.c
- Disk: Added the ability to queue file transfers.
- Disk: Combined the file operations delete, chmod, mkdir, and rename into a single thread and added file operation response messages to inform GUI clients of operations' results.
- Disk: Enhanced the read/walk packets by storing permissions as an unsigned int instead of a 12 byte string, fixed walk sig check issue, and increased performance of the database thread in the control.
- Disk: Fixed an issue where offline browsing large directories would improperly display the deleted display icon for the majority of the directory contents.
- Disk: Enhanced the file transfer capability by sending out a range of matching blocks (tokens) rather than individual packet for each matching block.

- Disk: Enhanced performance of disk read and walk operations so it consumes about 40% less cpu.

15.15.2 User Interface

- Transport: Updated the Mask field in the Add Route Dialog to be a spinner instead of a combo box.
- Disk: Fixed issue where extra characters were showing up in the Permissions column of Windows nodes for certain versions of Java.
- Device: Fixed an issue that caused new UDP link creation with the Link Panel to fail when a user specified local port number is supplied in the spin box.
- Horizon: Fixed how combo boxes render device names to ensure that even if the device name is too long, it doesn't cause the combo box to be pushed off screen.
- Device: Added new system id to GUI for vanguard target.
- Shell: Added a 15 second timer that when necessary replaces the stack address with device name in the Shell Info table.
- Disk: Fixed issue where during a sync the wrong file transfer row was being updated due to multiple matches.
- Shell: Added support for new busybox version of Windows shell sessions and enabled shell session history for Windows.
- Device: Removed lingering println when creating a link.
- Horizon: Updated custom comboboxes to match the default L&F and size.
- Device: Added local port support for links in the Link Panel.
- Disk: Fixed Start Directory Walk button so that it displays an error message instead of the walk dialog if all currently connected nodes cannot be walked.
- Disk: Removed outdated code from the Directory Read Dialog.
- Transport: Fixed the height of the status bars in the global Transport window (route and rule panels).
- Shell: Fixed issue where the Shell Execute table in the command column displays nonrenderable characters due to improperly decoded Microsoft Windows UTF16 strings.
- Disk: Added the ability to process queued file transfers.
- Disk: Updated the CGM definitions and added file operation response message handling, which writes log messages and toggles the Directory Window's progress bar to indicate file operations are pending.
- Disk: Added the ability to convert the permissions value found in read/walk requests to human readable strings.
- Disk: Fixed an issue where offline browsing large directories would improperly display the deleted display icon for the majority of the directory contents.
- Disk: Enhanced drag/drop file transfers by allowing a file to be dropped anywhere in the target table.
- Disk: Updated the size column in the directory table to display a human readable size.
- Disk: Fixed directory walk operation such that walk requests cannot target other Controls.

15.16 Release 2.2.1 (July 28, 2011)

15.16.1 Backend

Fixes:

- Packer: Updated the version print to properly display the stack version.
- Disk: Fixed a memory overrun when saving disk read or walk results for symbolic link directory entries.
- Disk: Fixed cleanup in file transfer download and added missing unlock on failed block ptr.
- Netroute: Fixed removal of protocol static routes for linux.
- Transport: Fixed bug with uninitialized bytes being left during system route entry looping in netroute API.
- Node: Fixed path issue when node is run from different directory then that of the CWD
- Mainboard: Fixed an issue on Solaris 8 when retrieving CPU info
- Disk: Fixed race condition where a signal to a thread could be missed.
- Device: Fixed catalog id mismatch for configured systems.
- Tools: Fixed endian issue regarding the CRC value being set in packer.
- Tools: Fixed endian conversion in print statement in packer.
- Tools: Fixed endian conversion (os_id) in packer when extracting an archive.
- Tools: Fixed build issue in packer when STACK_VERSION was node defined
- Tools: Fixed variable initialization bug in packer.c
- Desktop: Fixed strncmp()/strlen() bug in system.c
- Tools: Fixed Some minor issues in packer
- Fixed infinite loop in disk database thread on controls. Bug occurs when performing offline browsing on a directory where previous online browsing or directory walking could not retrieve the directory listing due to lacking file permissions.
- Tools: Fixed timer config option for device in packer
- Utils: Fixed valgrind warning, “Invalid write of size 1”, in linux mainboard.
- Desktop: Fixed the use of an uninitialized variable in fprintf() that causes junk to be displayed to a user.

Enhancements:

- Device: Added version to device and expanded it to utilize the full four bytes.
- Core: Changed the plugin timeout value from 5 to 10 seconds.
- Scripts: Enhanced the disk log level to provide messages up to the Fatal level.
- Platform: Added Mikrotik MIPSLE support.
- Tools: Enhanced the extract (-e) option in packer to provide user friendly information about the plugins in an archive.
- Node: Added os_id, mesh id and crc to main archive header
- Core: Added the ability to securely delete files with ASCII paths and file names.
- Tools: Fixed node payload container size to be calculated at build-time.
- Shell: Enhanced the posix execute feature to prevent the full path of a binary from showing up in ‘ps’.
- Transport: Deleted 2 log messages in database that were causing the transport.log to fill up with unnecessary messages.

15.16.2 User Interface

Fixes:

- Transport: Fixed bridge add dialog's verification checks so it will allow multiple bridge definitions for the same local port.
- Transport: Fixed issue where only one of multiple bridges currently enabled populate the Transport Bridge table when Horizon GUI is disconnected then reconnected.
- Transport: Fixed combo boxes in Bridge Add/Edit dialog box so that they now update their contents when nodes are added or removed from the network.
- Device: Added Mikrotik as model for MIPSLE experimental support in Horizon GUI.
- Disk: Fixed a problem where finalized directory walk rows were being reused in Directory Walk Table due to a node's plug-in being unloaded and reloaded.
- Disk: Fixed disk info dialog size to prevent file names with a majority of capital letters from running off the dialog.
- Disk: Fixed string length of Link Target in the Info Dialog to prevent the Close button from being pushed off screen.
- Disk: Fixed a race condition in the dir walk table where the database icon would never change to indicate a completed walk.
- Deleted use of global variable referencing for the bridge model.
- Transport: Fixed delete operation for routes and bridge tables where the wrong row was being deleted.
- Shell: Fixed how an Execute window is opened if it already exists and added the ability to set the path in the command text area.
- Shell: Fixed Create Session dialog to be modal and did minor code cleanup.

Enhancements:

- Device: Added the stack version in the Dashboard title.
- Device: Added Mikrotik as model for MIPSLE experimental support in Horizon GUI.
- Transport: Added the ability to edit disabled bridges.
- Disk: Added the ability to execute a file from a Directory Window.
- Device: Added ability to query if a device is a Control.

15.17 Release 2.2.0 (May 4, 2011)

15.17.1 Platform Support

- Linux - x64
- Microsoft Windows 2008R2 x64
- Backfire - armbe

15.17.2 Core

- Memory management library supporting platform independent allocation and free calls.
- Protocol handler framework within the mesh networking stack.
- Version support for the mesh networking stack. Desktops and nodes are prevented from communicating across different versions.
- Fixed stack issue associated with tcp links operating on latent and
- UDP link timeouts for connected links are automatically increased for both minimum and maximum ranges.
- Fixed stack issue for incorrect timestamp differentials for acknowledged packets and expanded timestamp field to accommodate highly latent links.
- Support for HTTP protocol communications across links established in the mesh networking stack.

15.17.3 Device Module

- Unique identifiers per device generated on the fly using hardware tokens derived from the mainboard and netiface plugins.

15.17.4 Disk Module

- Enhanced file transfers using a custom delta encoding algorithm. Assists in faster data sync and resuming file transfers that were interrupted.
- File system directory walk and database storage of the results.
- Online/Offline directory read/walk results in Horizon.
- Make/Remove directory.
- Rename of file or directory.

15.17.5 Horizon

- Column sort feature enabled on necessary tables.
- Copy/Paste capability of the AH key in the Device Status panel.
- Made Log Window's options allow category and log level to filter messages.
- Unified look and feel of all tables.

15.17.6 Utilities

- Mainboard plugin supporting lookups of CPU, RAM and PCI information.
- Netiface plugin supporting lookups of network information.
- MD5 plugin supporting block, file, and hmac capabilities.
- Base64 plugin supporting encode and decode of binary data to ASCII string.

15.18 Release 2.1.3 (April 15, 2011)

15.18.1 Transport Module

- Fixed session reset issue with nodes when connections spread across the available runners.

15.18.2 Horizon

- Updated Transport Status panel to display the number of active TCP/UDP connection along with Accept/Drop packet counts.

15.19 Release 2.1.2 (March 15, 2011)

15.19.1 Core

- Added poll support to the network library with emulated support using select system calls.
- Optimized stack event loop using the internal poll support available in the network library.

15.19.2 Transport Module

- Optimized the session loop using the internal poll support available in the network library.
- Fixed local stack to properly support nmap scans using TCP or UDP.

15.20 Release 2.1.1 (Febuary 15, 2011)

15.20.1 Platform Support

- Mikrotik - mipsbe and powerpc

15.21 Release 2.1.0 (January 12, 2011)

15.21.1 Platform Support

- Linux - x86
- Solaris - x86 and sparc
- Microsoft Windows 2000, XP, 2003, 2003 R2, Vista, 2008, and 7 - x86 and x64
- Mikrotik x86
- Paradyne mipsle (experimental)

15.21.2 Core

- Plugin architecture using either memory (pom) or disk (pod) resident formats. Memory resident plugins (poms) are loaded and executed in memory while disk resident plugins (pods) are loaded from disk and executed in memory.
- Plugin manager to catalog plugins, descriptors, interfaces and etc.
- Overlay manager to handle I/O of packets between the stack and modules including module to module messaging.
- Archive format to bundle one or more plugins and their dependencies and configurations into one blob.
- Stack supports mesh layer reliability ensuring packet delivery between distant nodes regardless of network disturbances.
- Optimized mesh network stack processing algorithm to maximize packet delivery and response.
- Support for module interfaces using both static and queued methodologies. Primary mechanism for utility plugins that expose their interfaces to modules.
- Limited diagnostics for the mesh networking stack. Route tables for each device can be pulled back for troubleshoot discovery.

15.21.3 Packer

- Generate custom nodes with the following features; 1) Plugins and their configurations, 2) OS, CPU, model, and version, 3) Authentication code, 4) Device name, 5) Multiple device service ports, 6) Multiple device link configurations, 7) Maximum transmission unit, and 8) Timeout with delete option.
- Nodes can be deployed using a one integrated executable or one lightweight executable combined with compressed archive file.
- Advanced help system to display available systems and plugins to operators. This also entailed developing stronger checks when a system is chosen to ensure the required files exist before building a node.

15.21.4 Device Module

- Plugins can be manually loaded and unloaded through the user interface. Plugin dependencies are automatically generated and processed by the plugin manager.
- Device timer enhanced with a smart timeout and interval algorithm, minimizing shutdown on congested networks. If at minimum one keep-alive is received during the timeout interval, the device will complete that interval and proceed to the next timeout interval. Keep-alives intervals are specified by the operator and define the period of time to broadcast for a known control device on the network.
- Process identifier (pid) as represented by the platform resources for both controls and nodes is displayed in the user interface.

15.21.5 Transport Module

- Node utilizes a throttle algorithm to minimize saturation of the mesh network and reduce starvation of new sessions.
- Node virtual stack incorporates highly scalable threading model to support 512 (mswin) and 2048 (posix) concurrent sessions.

- Node virtual stack can process multiple connect messages without any delay using an asynchronous TCP connection thread.
- Control virtual stack supports accurate TCP handshake simulation. Handshake will be evaluated at the gateway node before allowing the control's stack to proceed. Scanning tools configured for transport layer surveys (nmap connect scan) will report accurate network information.
- UDP timeouts are configurable through the desktop configuration file or upon configuration of a UDP bridge.
- Bridges are synchronized across the mesh network. Operators will see realtime listing of all bridges utilized in the network.

15.21.6 Disk Module

- Incorporated a file transfer throttle algorithm to minimize CPU spikes and network congestion during large file transfers or during maximum concurrent transfers.
- All active file transfers on the network are synced with the control on connect to provide an operator with an accurate view of current activity.
- Operators can register/unregister with file transfers that do not belong to them to receive periodic updates while they are active.
- Files with matching names are no longer overwritten. File names are incremented and saved in the requesting directory.
- File transfer stats are limited to prevent overwhelming controls. Stats packets are sent on 1% boundaries with a maximum of 100 stats packets per file transfer per control.
- Optimized the speed at which directory listing request/responses are handled and processed.
- Added the ability to drag/drop files from one directory listing to another to initiate new file transfers.
- New confirmation dialog is shown to an operator before files are actually deleted.
- File transfer history provides a status field which displays the current status of a given file transfer (i.e., File transfer cancelled, File size 0, File transfer completed, etc.).
- Ability to change/update a file's permissions and other special bits.
- Copy Path feature which provides the ability to copy a file's path (shown in a Directory Window) directly to the system clipboard for later use.

15.21.7 Shell Module

- Built and integrated a command line client terminal directly into Horizon.
- Nodes use native command line facilities for each target operating system (i.e., Unix prompts look like Unix and Windows looks like Windows).
- Incorporated the ability to Copy and Paste text throughout shell sessions. Users can Copy and Paste between shell sessions, other Horizon components (i.e., Disk Module), or external applications.
- Ability to highlight and export selections to a text file from the shell session.
- Internationalized character support for Latin and Asian languages with appropriate Unicode encoding for the target operating system.
- Shell Session windows provide a font selection dialog to allow an operator to select a different font if the current font does not include characters for the language currently displayed.
- Terminal spaces double width Asian symbols correctly and advances the cursor properly.

- Terminal supports interactive console programs like vim, emacs, and edit.
- Common key commands like Ctrl+C and Alt+F are handled appropriately.
- Shell Session windows include a scrollable pane with a scrollback buffer for Unix nodes.
- All sessions are synced with the Control to provide the operator with an accurate view of sessions spawned throughout the mesh network. Any control can spawn, attach, detach, and terminate a session.
- Operators can share individual sessions between multiple controls to allow collaboration within individual sessions.
- A quick batch execution feature allows operators to configure and launch individual executables without needing to spawn a full shell session.
- Horizon provides a table to show the results and history of batch execution.
- Added a secondary execute feature that provides an operator with a command line like interface to execute binaries on a remote node. Operators can enter Environment variables, binary path, and arguments.

15.21.8 Graphical User Interface

- Redesigned graphical interface architecture to support modular window scheme.
- Network listing windows provides convenient access and display of data including realtime round trip time and packet loss updates for each device.
- Custom dashboard that provides critical information such as client status, # of unread messages, and Horizon uptime. Module (Transport, Shell, and Disk) specific information is also provided to allow an operator to quickly find information relating to each module loaded on the control.
- The module panels in the dashboard are dynamically enabled/disabled based on whether that module is loaded on the connected control.
- Dashboard provides access to all of the global windows (system and module related) as well as a dynamic list of all open and active windows. Operators can select a window from the list of open windows to quickly bring it to the front of all other windows.
- The dashboard can be pinned to ensure that it remains the topmost window at all times. This feature can also be disabled at the operator's discretion.
- Advanced GUI logging feature with custom sort algorithm to provide only the required information to operators. The Log window provides the following information: Read Status, Timestamp, Level, Category, and Message and can be sorted by Level or Category.
- All log messages that are of Level: Error immediately brings the Logs window to the front and increment the number of unread messages. As each of these messages are read by an operator the unread message count is decremented.
- Rolling log files are created in the GUI to provide a backup copy of what is presented in the log window.